

QL Today

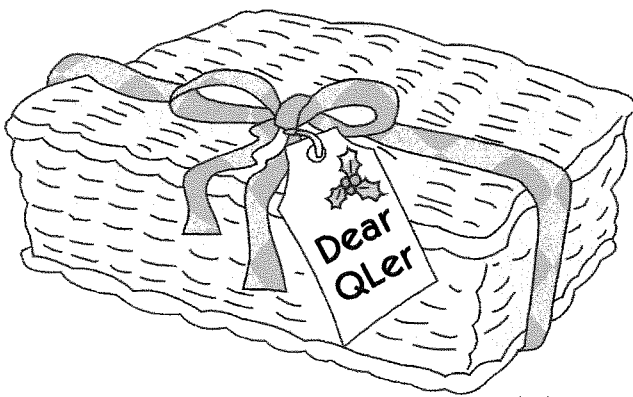
Volume 6
Issue 4
Nov/Dezember
2001

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

www.qltoday.com

A NICE SURPRISE FROM SANTA QLAUSE?



Contents

- 3 Editorial
4 News
7 Small Ads
7 Anagram Replies
8 Introducing QDT: A Progress Report
Jim Hunkins
15 Gee Graphics! (on the QL?) - Part 25
H L Schaaf
16 Two Users
Geoff Wicks
17 A new QL Publication
Georgio Garabello
18 Using INdexFindString with DATAdesign
Wolfgang Lenerz
23 German/Austrian QL Show in
Berchtesgaden
Jochen Merz
25 Error Trapping comments
26 Letter-Box
34 Programming in Assembler - Part 9 / 2
Norman Dunbar
43 Notes on Norman Dunbar's Assembler
Parts 8 & 9
George Gwilt
46 The DEMO VERSIONS - Cover Disk 2
Dilwyn Jones
50 Byts of Wood

Advertisers

in alphabetical order

Dilwyn Jones	55
Jochen Merz Software	13
QBOX USA	23
QBranch	42, 43
Quanta	31
RWAP Software	45
W.N. Richardson (EEC)	27
TF Services	19
Geoff Wicks	7

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg Box1 +49 203 502013
Germany Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@J-M-S.com
email: jochen@QLToday.com

English office:

Q Branch Tel. +44 1273 386030
20 Locks Hill Mobile +44 7836 745501
Portslade Fax +44 1273 381577
BN41 2LB email: qbranch@qbranch.demon.co.uk
United Kingdom

Editor:

Dilwyn Jones Tel. +44 1248 354023
41 Bro Emrys Fax +44 1248 354023
Tal-Y-Bont, Bangor email: dilwyn.jones@dj.softnet.co.uk
Gwynedd email: dilwyn@QLToday.com
United Kingdom LL57 3YT

Co-Editor:

Bruce Nicholls Tel +44 1708 510764
38 Derham Gardens Fax +44 870 0568755
Upminster email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA email: bruce@QLToday.com
United Kingdom

Please send comments and articles to
editor@QLToday.com

QL Today is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

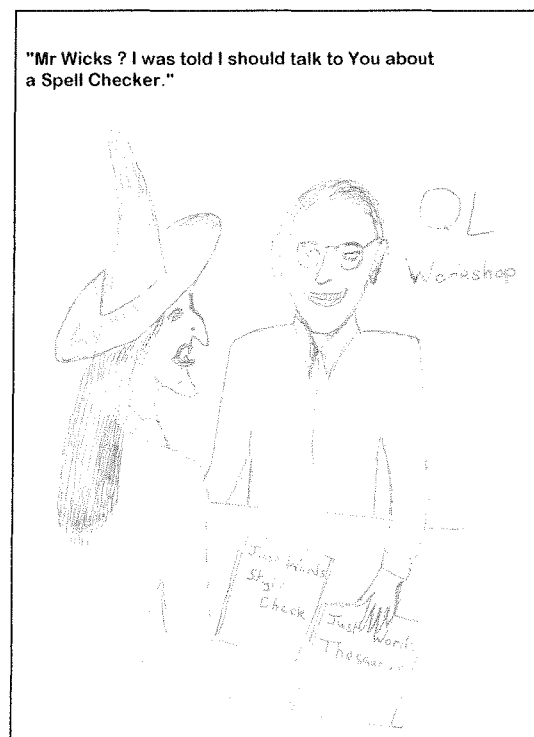
QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2000 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

First of all, let me apologise if this issue gets to you later than it should. Jochen has been unwell and I have been caught between changing jobs and work on an overrunning extension building project at home – everything has been in boxes for months here and it is only now, some 2 months late, that things are beginning to return to normal. Rumours (good news if true!) from the Q60 department: Peter Graf has told me that the Q60 is to go into production in the not too distant future, but he is not yet ready to say who will be producing it. Hopefully, more details to follow in the next issue.

What with the arrival of the 'colour drivers' (or GD2 to use the proper name), we dragging the QL and compatibles into the 21st century. The three things we have been largely missing have been proper CD ROM drivers (largely solved by Thierry Godefroy's system for Qubide and Q40/Q60), internet access (addressed by Jonathan Dent's soql system still unfinished) and a proper desktop GUI system to tie together the various parts of the pointer environment and the various useful utilities available out there. So far, there had been program launcher systems like Jonathan Hudson's Qascade system (similar to the Windows Start button) and my own Launchpad system (more news on that in the next issue) but finally we have QDT, the QL Desktop from Jim Hunkins. Jim works for Apple as a systems designer, so has plenty of experience with these things, and in this issue he details his work on the QDT system expected to be available sometime next year. As you'll see from the article, work is well advanced. That, and the completed soql, will be high on my QL wish-list for 2002!

I'd like to thank all concerned with QL Today for their work and help during the year. Despite being bi-monthly, QL Today is a lot of work for Jochen and I and we are very grateful to people like Bruce and Roy who do a lot of quiet behind the scenes work for us (Bruce does the proof reading for example), and of course to our regular contributors without whom there's no magazine of course. Keep up the good work everyone so that we can keep QL Today as your one-stop source of QL News and Information. And of course, we always appreciate new contributions – if what you'd like to see in QL Today isn't there, write an article for us, every time we see a new name appear in QL Today, more people write for us, so however short and trivial you might think your contribution is it all helps! Finally, enclosed is the second of our 'Demo Version' cover disks, packed with demonstration versions of commercial QL software. Enjoy.



Cartoon

NEWS

Jochen Merz Software News

QPC2 Version 3 is ready! Marcel has implemented a vast range of user-requests in the very latest version of QPC2. Here an extract of the new features. First some minor, but very useful things:

- Resize with locked aspect ratio. This ensures that the x/y ratio will remain fixed if you manually resize the QPC window.
- It is possible to activate a power-save mode. This should be extremely useful for battery powered laptops, when QPC is idling.
- QPC2 now starts centered on screen if run in window mode.

Now the real add-ons:

- PAR2, PAR3 and PAR4 have been added, which means, you can have up to 4 real printers (anywhere) or printers connected to LPT. The PAR and SER drivers have been improved to take care of strange behaviours of "Windows print managers". If you experienced problems in the past, they should be gone now.
- It is possible to supply the base directory for any of the DOS devices. This means, a path can be provided like C:\myfiles\ or \\SERVER\SHARE\
- Wheel mice supported (from Win98 on).
- QPC now emulates the original 512x256 QL screen at \$20000 in all resolutions and colour depths. This means, that old programs which write directly to the old QL screen area will work again and display their screen output properly.

Other useful control features are:

- QPC_NETNAME\$ is an SBASIC function which returns the network name of the current PC. This allows you to distinguish between several PCs in a BOOT program, for example. Very useful, if you use QPC on more than one machine!
- You can control the QPC window with SBASIC-Functions, which allow you to minimize, maximize or restore the window.
- It is possible to start external windows programs or files, e.g. "QPC_EXEC 'notepad', 'c:\text.txt'" or "QPC_EXEC 'c:\text.txt'" to start the default viewer for txt-files.

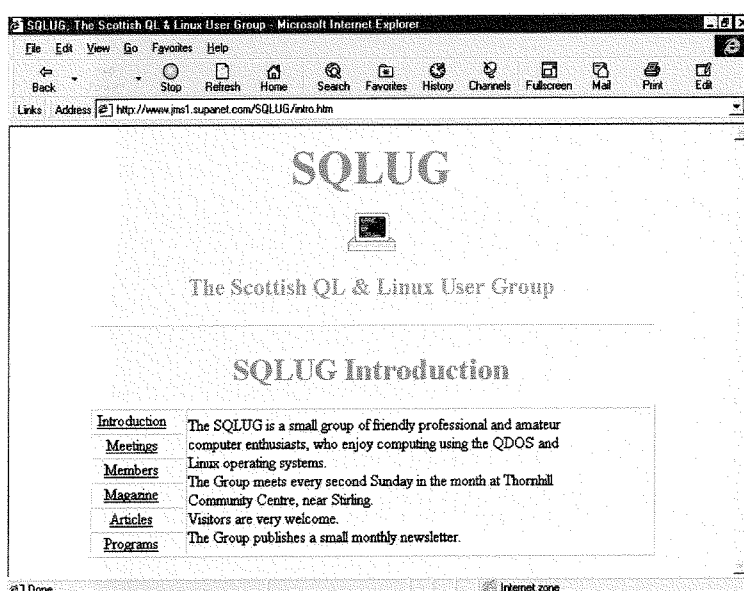
The upgrade from QPC2 Version 2 to Version 3 costs only EUR 20. A new manual for the QPC2 part is included as well. Please return your master disk by post. Please note: from now on, QPC1 will not be included on the disk anymore! If you want to continue to use QPC1, make yourself a copy of the master disk before you return it! A very special offer to QPC2 Version 2 users who upgraded or bought QPC2 Version 2 recently: if you bought it after September, 15th, you'll get the upgrade for only EUR 6 to cover copying costs. All prices plus p&p, of course. This special offer includes all sales and upgrades from the recent computer shows (Byfleet, Berchtesgaden and Eindhoven).

SQLUG NEWS from John Sadler

SQLUG, the Scottish QL and Linux User Group, has started up a small web site at

<http://www.jms1.supanet.com/SQLUG/intro.htm>

It is still very much under development but I hope in due course it will be of interest to readers of QLtoday.



Linux on Q60/Q40

For those who use the Q40-Linux on a Q40 or Q60 QL-compatible, Thierry Godefroy maintains a site at q60linux.free.fr

At the present time, you will mostly find some pre-compiled RPMs as well as some useful links to other Q60/Q40 sites.

Some links to related (Q40/Q60 and Linux-68k) Web sites:

<http://www.q40.de/>

Q40/Q60 main site by Claus & Peter Graf.

<http://www.geocities.com/SiliconValley/Bay/2602/q40.html>

Richard Zidlicky's Linux-Q40 page.

<http://www.geocities.com/SiliconValley/Bay/2602/m68k.html>

Richard Zidlicky's Linux-68k page (kernel patches).

<http://sourceforge.net/projects/linux-q40/>
Linux Q40 distribution on sourceforge.

TURBO NEWS

The latest Turbo Compiler v4 release 12 (v4.12), Turbo Toolkit v3.31 and TurboPTR v3.6 from George Gwilt is now available. Also included is the latest version of the Demos_bas files (v3.04) and the updated extra utility tasks from the Turbo package (Library Manager, Utility Task, and Data-space Task) which are in the compiler package. Changes to the compiler itself in v4.11 and v4.12 include:

v4.11

1. Since v4.3 it was not possible to use WHEN_ERROR 1. This has now been corrected.
2. A WHEN_ERROR clause can legitimately be written without a RETRY. This would not work unless a RETRY_HERE had been issued. In v4.11, a RETRY_HERE is automatically set at the start of every program.

v4.12

1. The implementation of WHEN_ERRORS, wrong in previous versions has now been corrected.
2. Floating point division by zero now always produces an overflow error. Previously zero divided by zero gave the answer 1.
3. A correction has been made to the floating point division routine in some cases when an overflow occurs. This enhanced floating point division relies partly on using the results of a previous calculation, which therefore does not need to be repeated. However, when a QDOS overflow error occurs the results of the division cannot be properly stored. If this condition was not trapped the consequence would be that on a following division an erroneous answer could be given and no error signalled. This circumstance could arise if the overflow is dealt with by a WHEN_ERROR clause which includes a RETRY.
4. BLOCK can now be used with the colour driver GD2. Previously all BLOCK commands were interpreted ignoring any COLOUR_PAL, _24 or _NATIVE directives which may have been issued. This resulted in the block colour being interpreted as if in mode 4 or mode 8. If the colour parameter was less than 32768, the block would be printed, but with the wrong colour, otherwise an overflow would be reported.

The latest programs can be downloaded from the Other Software Page on the Dilwyn Jones website:

<http://www.soft.net.uk/dj/software/other/other.html>

PSION CHESS FOR DOS

I came across a freeware version of Richard Lang's QL Chess (you probably know it as Psion Chess) which runs in DOS or Windows. For nostalgia's sake, and in case anyone would like to try it against the QL version (which as far as I know is not freely distributable) it has been placed on the Other Software Page on my website.

<http://www.soft.net.uk/dj/software/other/other.html>

For information, Malcolm Lear has written a QL-QPC screen transfer program which seems to work by copying screen data from the old QL screen area which still seems to be available on QPC, to the current screen, but only works in the 4 colour screen mode. Since it needs a patched version of Psion Chess to work, it doesn't seem worth offering the routine as such, though it may be worth pestering the emulator authors to include such code in the emulators to increase compatibility with old programs which naughtily write to the old 32K QL screen memory. If memory serves me right, the Aurora already seems to do this in the sense that although you are in a high resolution screen mode, anything you write to the old screen area seems to get copied to the new screen areas after a very short delay.

Marcel has implemented this in QPC2 Version 3.

FREDDY VACHHA

Long standing QLers who remember Freddy Vachha of Digital Precision Ltd may like to know that I had the pleasure of meeting him a few weeks ago while I was working at Snowdon Railway over the summer and autumn period. We had a brief chat while he was waiting for a train. Although he would hardly describe himself as an active QLer these days, he did say quite plainly that he still has a QL and Gold Card, and still uses Perfection which he described as "less frustrating" than many word processors he has used. He said he was unaware of QL2000 last year, or at least if anyone had mentioned it he'd forgotten about it, and may well have attended had he known. He is open to invitations to such future events, as he still has a soft spot for the QL, despite not using his that often these days in view of the need to use other computers for his work.

Those who know Freddy will be glad to know that he is still his usual happy, bouncy self, happily

making money (forgotten what he said he was up to and where he lived now, should have written it all down and used it all in evidence!) and still has a soft spot for the QL.

NEWS FROM NORMAN DUNBAR

I'm in the process of moving my website and I have a new email address.

<http://www.bountiful.demon.co.uk/qdos>

It is blank at the moment but that's where it will be eventually.

My new email address:

Norman@bountiful.demon.co.uk

In case anyone is wondering, the 'bountiful' part is my wife's company - dunbar was already in use. We also now have www.bountifulsolutions.co.uk, but it simply redirects you to the demon web site.

QUANTUM WEBRING NEWS

The company which was behind the Quantum Ring linked list of QL-related websites has now been devolved from internet giant Yahoo. They recently sent an email to all list members advising them to request that their sites be transferred to the new WebRing servers.

Yahoo's "Terms of Service" does not allow member information to be transferred to another entity, including WebRing.

Please note, your ring/site WILL NOT be transferred automatically, you must follow the links given in the email and complete the transfer process. Rings and Sites not transferred will be lost once the migration period is over.

Questions can be best answered by visiting the new <http://dir.webring.com/rw>.

QLers whose sites are part of the Quantum Ring, please remember to request transfer of your sites so that we may keep the Quantum Ring going with all member sites intact!

PROWESS NEWS

Well, not news really, more of a reminder as it has been available for a while. Prowess users can download the 'colour drivers' update to Prowess from Joachim van der Auwera's website:

www.triathlon98.com/PROGS

This makes Prowess useable in the high colour modes of systems like QPC2 and Q40/60.

JUST WORDS NEWS

Geoff Wicks

QL-RHYMES has now been released. This is a rhyming dictionary containing almost 49,000 words that can be searched either phonetically or by end letters. QL-RHYMES is supplied with a

3,000 word laser printed manual. It costs £10 or 15 Euros.

[QL Today hopes to bring you a review of this program in the next issue - Editor]

DJ PD LIBRARY

Dilwyn Jones

A PD Software Library catalogue is now available for download or online viewing on the Dilwyn Jones website. At the time of writing, the library consisted of 34 general disks of free QL software, 15 disks of free QL games, 5 disks of QL-related text information (manuals, articles etc), 46 clipart packages, a few literature disks including a text encyclopedia and CIA World Factbook 1998, and some 26 disks of demonstration versions of commercial QL programs. Any QLer who wants a copy of my catalogue and does not have internet access facilities can of course send me a disk (DD or HD) to get a copy of the catalogue - see the address in my advert in QL Today.

The entire PD library is available on a CD for only £10.00 if you have CD-ROM (QXLWIN) access. This may well be useful for anyone who would like to set up their own PD library for example. Contributions of software from freeware/PD/shareware etc authors to the library are most welcome. This library is non-profit-making and being done solely to help spread QL software and the QL gospel generally.

The catalogue may be accessed via this website address if you have internet access:

<http://www.soft.net.uk/dj/djpdsoft/index.html>

Q-CELT COMPUTING NEWS

All been rather quiet of late due to the pressures of work at the Bank of Ireland where I have been working on getting the computers ready to handle the Euro in 2002. I am changing jobs soon (becoming involved with a new IT Centre in a village close to where I live), so the long hours I have been working this year may hopefully improve!

The QL Emulators CD is now at version 1.18 which includes new versions of the Turbo Compiler and a few other programs. The CD is now being produced in a combined Joliet and Rockridge format which should make it more useable for Amiga and Linux users, although the Rockridge part is largely experimental as Dilwyn is using an alpha-version of a Windows port of a Linux package to produce it, and like all alpha-version software it is still subject to testing (volunteers?)

The Phoebus Dokos Interactive Fantasy Collection CD of adventure game type material is now available along with the Zexcel Spectrum Emulator

CD which includes a large amount of Spectrum programs for use with the emulator. We are still working on the ZX81 Emulator CD and a CD from Phoebus Dokos and Dilwyn Jones of new fonts for use with LineDesign, Prowess, Proforma etc. Release dates for these are not yet certain.

A new CD of Line Design Clipart called Famous Faces, containing cartoon or caricature style pictures of famous persons past and present (sporting figures, historical, political etc) from Dilwyn Jones is now available - all the clipart on this CD is freeware.

The DJ Bargain Bundle CD is also due for updating to include the various new programs Dilwyn has written in the last few months - keep on eye on our adverts and news releases for details.

QL Roadmaps: What piece or pieces of hardware or software changed the way you have used your QDOS/SMSQ system over the years? We would like to hear from users saying when they bought their QLs, and when these signposts or revelations happened. You can keep the replies as short as you want but we feel that other users may benefit from your personal epiphanies.

Roy Wood

Small Ads

For Sale:

3 x Syquest 135mb EZDrive ... £40 each

12 x Syquest 135mb Disks ... £7.50 each

Price includes postage and packing,

can be picked up in the Codnor/Derby Area

Please telephone Derek Stewart after 18:00 on

Tel . 01773-741164

Anagram Replies

Stephen Poole writes:

Regarding your challenge on page 37 of Vol 6 Issue 3, do you remember PRIDE IN ITS LOGIC (the advert anagram of DIGITAL PRECISION)? Well, this is my offering:

Psion Italic Dirge

Dare I, In IPC List, Go

Indigo Plastic Ire!

I Instil A GC Period

Citing Idol Praise

It Is A Peril, Doing 'C'

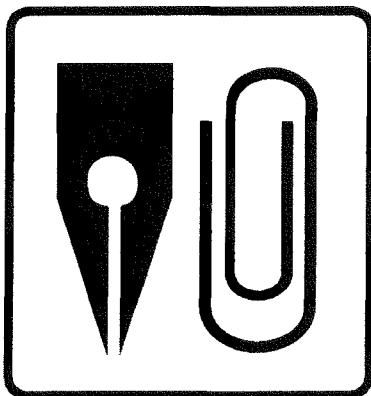
Raiding Its Police?

In Prelogistic Aid

It Is A Price In Gold

by Realistic Pongidi

Yes, they are all anagrams of Digital Precision.



QL-2-PC TRANSFER

The ultimate file transfer utility. Converts Quill, Perfection and, to a lesser extent, Text87 documents to RTF, HTM or ASCII formats. Converts HTML files to ASCII. Converts QL character set to Windows or DOS codes and vice versa. Removes unwanted spaces, line feeds and soft hyphens from OCR read or downloaded texts. Extracts the text message from emails.

£10 - €15

QL-THESAURUS

23,000 words divided into 1,000 groups.

STYLE-CHECK

Check and improve your documents.

QL-RHYMES

Phonetic and End Letter searches.

Geoff Wicks, 28 Ravensdale, Basildon, Essex, SS16 5HU, UK.

Tel: +44 (0)1268 281 826

Email: geoffwicks@hotmail.com

Web: <http://members.tripod.co.uk/geoffwicks/justwords.htm>

Just Words! - Software for Writers and Word Lovers.

Introducing QDT: A Progress Report

Jim Hunkins

This last summer I had the honor of introducing my current Software project to the attendees at the North American QL Show. This was the first time that anyone except for two people had seen QDT: the QL DeskTop. For those who have not seen this other than for the short notes in QL Today, this article will introduce the project and give a progress report.

Background

Anyone familiar with one of the 'modern' computers such as a PC or Macintosh understands the idea of a desktop. Simply put, it is a structure to help organize your files, launch programs, and use a computer more efficiently. To help you picture the concept, first think of your physical desktop at home where you keep your papers, folders of bills, a favorite picture or two, and tools such as a calculator, pens and pencils, etc. Now move this idea to your computer. You have a note or two displayed all the time on your screen, some folders which, when opened, contain documents of some type, another folder which may contain your programs, while another yet may contain your calculator program or your notepad program. And of course, to display your pictures of your significant other (or in my case my two cats), you have the pictures on your computer desktop. Currently on the QL we have bits and pieces of this [Figure 1]. But unfortunately, they are not tied together so that your computer can really work and feel like an efficient desktop.

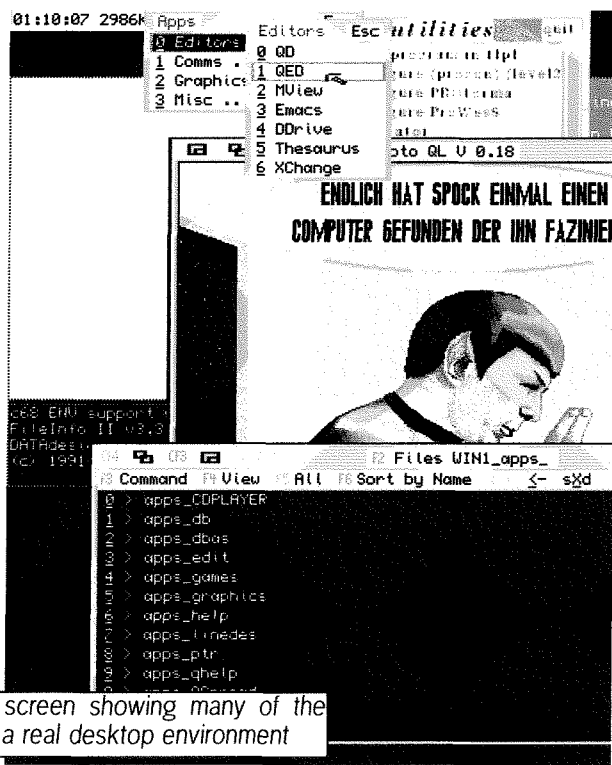


Figure 1: A QL screen showing many of the components for a real desktop environment

users have the following:

Windowing Environments:

- Pointer Environment - tools, pieces, and framework
- ProWesS

Components

- Disk Directories
- Hot Keys
- Buttons - normal and menu
- Q-Menu
- File Associations

Utilities

- File Managers
- File Viewers - text and graphics

In the world of PCs and Macs, you find operating systems with built-in desktops such as Windows 3.1, 98, or XP, Macintosh 9 or X, OS/2, Solaris/X Windows, etc [Figure 2]. These all include very comprehensive desktop features and capabilities.

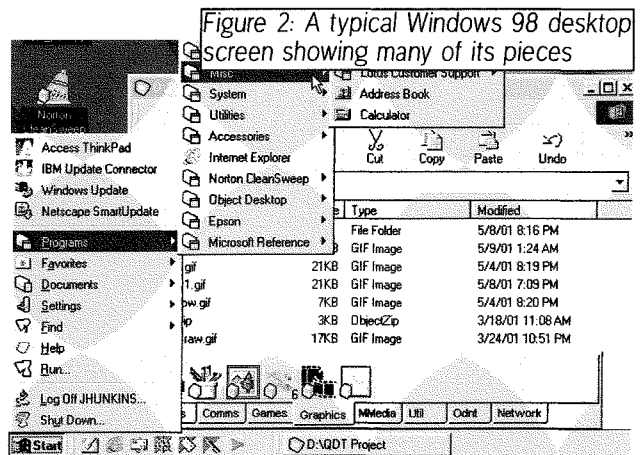


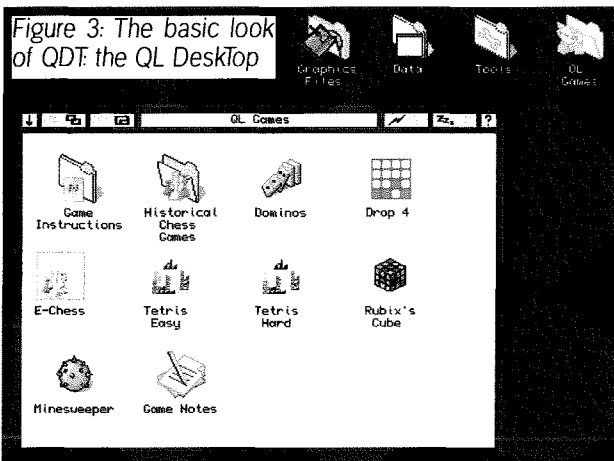
Figure 2: A typical Windows 98 desktop screen showing many of its pieces

But of course they come with a price. I personally find the three worst costs associated with them being:

- 1) the sheer size and complexity of the code
- 2) the overhead to run the large code base
- 3) the difficulty to program for them

And Now For a High Level View of QDT

The answer to this is a piece of software that adds functionality to the current QL capabilities. It is software that reuses pieces where they make sense and adds new pieces to tie it all together into a neat, useful, and graphically pleasing desktop experience for the QL user. I would like to introduce you to a first look at QDT: the QL DeskTop [Figure 3]. By the way, please use your imagination for



the color. The original image is actually using the newer 256 color mode and is very cool looking on a computer screen. In this illustration you see the basic pieces of QDT. Before I explain them, let me state that QDT is based on objects. Everything on the desktop is an Object. In Figure 3, you see four folder objects along the right hand top (commonly called icons – a small image). Three are closed folders and one is an open folder (notice how the icon shows what looks like an actual paper folder that is open, just like a physical desktop would have).

Since you see the open folder object, you would expect to see something showing the contents of that folder. And you do in the larger window that is the opened folder object itself. Within this folder, you see several objects represented also by icons. In this case the different objects displayed represent additional folders (open and closed), program objects, and even a text file object. The folder objects can be opened and closed just like the folder they are in (notice that you can put folders within folders, just like in real life. But don't get carried away, just like you wouldn't in real life; who wants a 100 pound folder to search through!). The program objects launch the programs that they represent. And the text file object normally would launch some type of editor/viewer to read and/or edit the text file. There are several other types of objects that you can have which I will discuss later in this article. What this simple screen capture (yes it is generated from real QDT code) represents is the set of basic building blocks of the QDT desktop. All the other pieces support this structure, either by adding functionality to it or by making it easier to work with it through different tools.

QDT Pieces – Programs and Utilities

In addition to the 'master' QDT program, QDT comes with an array of other programs and utilities. It also takes advantage of several

shareware and commercial programs already available to enhance and expand its capabilities and usefulness. Here I will outline the current pieces in or soon to be in development.

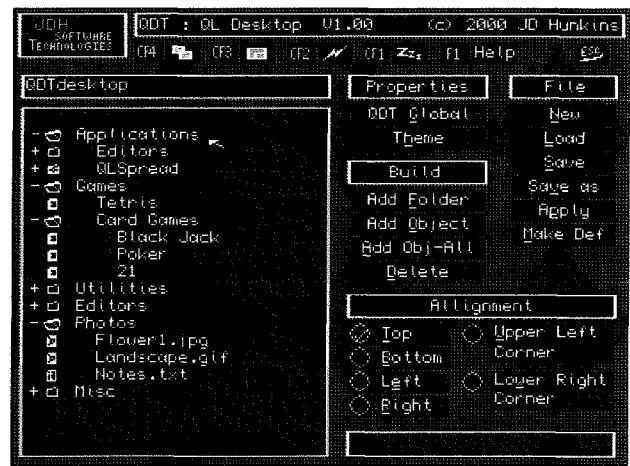


Figure 4: The Desktop Builder tool

Desktop Builder Tool [Figure 4]: while you can change and build your desktop directly from within the desktop, some users may find it easier and/or desirable to do some of the work with this utility. The Desktop Builder can build a default desktop automatically, change between different desktops (for different users or different functionality such as book writing, code development, etc.), or search and add your drives for all objects in its database of common QL programs. It also allows the user to manually add or remove object folders, programs, and files.

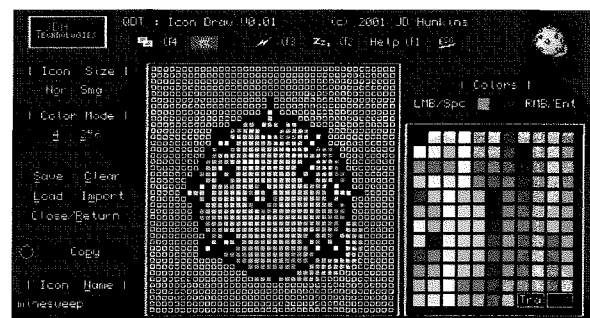


Figure 5: IconDraw tool

IconDraw Utility [Figure 5] : this utility is simply fun to use. Part of any desktop is the look. And the icons make the look. With the new color drivers for the QL systems, you can have some very cool icons. This program lets you draw new icons, load and change existing icons, and to even import icons from other systems such as Windows (there are thousands available for free). This was the first program developed for QDT and was very useful in learning all the basic things about programming in 'C'. It also helped to lay a lot of the database groundwork for the project.



Figure 6 : Default Icon Manager

Default Icon Manager [Figure 6] : this program was first developed to build the default icon file (for objects that the users does not specify an icon for). Realizing that QDT users will likely be like everyone else and not be happy with things as they are, the utility was expanded and made available to change the default icons to what ever the QDT user wants them to be.

Theme Manager : this little utility will allow the user to pick from one of several pre-designed desktop themes (color schemes, default icons, etc). Hopefully users will develop and share different desktop themes amongst themselves.

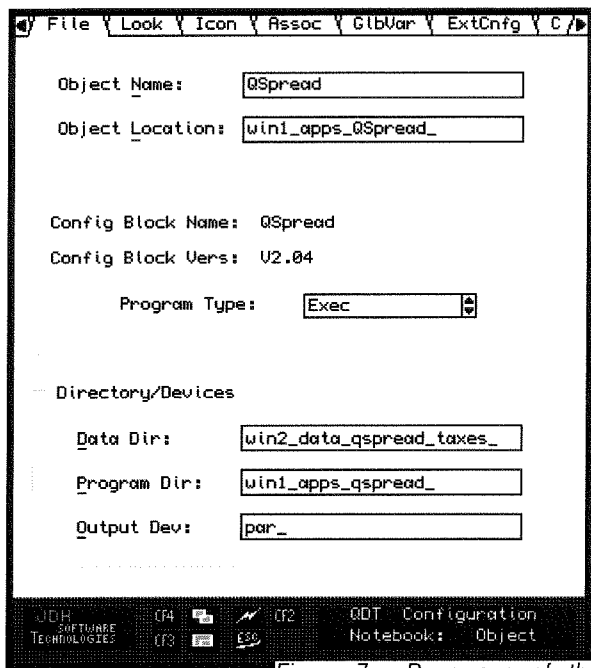


Figure 7 : Page one of the Object Configuration Notebook

Tabbed Config Notebooks [Figure 7] : One of the items in QDT that will help bring the QL up to current computer standards is the tabbed configuration notebook. The notebooks introduce tabbed pages (like an index file). They also add many new control features beyond what the pointer environment includes such as scrolling and dropdown lists and button types. In addition to pages for QDT required options, the plan is to

incorporate pages for the standard config blocks built into many programs and for separate config files for specific programs. With these additional options, they will be useful for other programs to use and can even be used to replace the current config block configuration programs.

Installer : One of the most common complaints about the pointer environment is how hard it is to initially set up and to use. ProWesS went a long ways towards developing a good installer system which helps the user get past the initial learning/resistance curve. QDT will attempt to go one further. The QDT installer will allow for different levels of user interaction during the installation process. The current thinking is to provide both an EASY (fully automatic) and a INTERACTIVE (full user interaction at each step) installation process. The installer will be able to modify the boot file, search the drive for programs, automatically build a desktop, and link in third party programs.

Job Manager : QDT heavily uses the multi-tasking capabilities of the QL operating system. Every folder and every desktop object has its own job, along with separate jobs for drop down menus and warning/error windows. This can result in a huge list of jobs running when using the QPtr jobs function. QDT will provide its own job manager which can be set to filter out different types of jobs. This manager will also have options such as shrink, open, and kill.

Tab LaunchPad : Okay, now that everyone has been sold on the idea of having folder objects to store program and file objects in, it is time to look at an alternative way to access commonly used programs and objects. What ends up happening with many user's desktops is that some of their objects can get buried down in folders, slowing the access to them. For objects that are accessed all the time, it is handy to have them placed for quick access. Some users of other systems often try to put them on their actual desktop, which is perfectly legal. However, if you store too much there, the desktop gets cluttered which makes it hard to find things (not to mention, it looks just messy). The Tab Launchpad is simply a special type of folder object that is in the shape of a long box holding objects tightly packed next to each other. It has different tab pages just as the configuration notebook, allowing the user to sort objects by category. By holding the cursor over an object, the objects title appears temporarily.

Integrated Cross System Help : There are at least three different help systems currently available for the QL. However, they are for the most part standalone. The attempt within QDT will be to tie a help system into the desktop itself so that it can be called up from just about anywhere and display the appropriate related information. At this time it has not been decided whether to use a currently existing system for the core or to develop one from scratch.

QDT Pieces – Objects

As has been discussed, objects are the core of the QDT design. Each object has different responses to the left or right mouse button (LMB or RMB – same as space and enter from the keyboard) along with

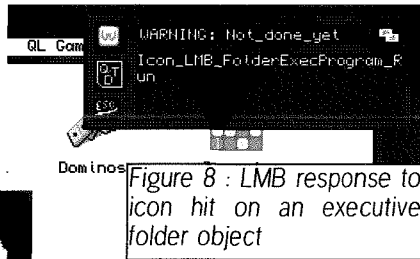


Figure 8 : LMB response to icon hit on an executive folder object



Figure 9 : RMB response to text hit on a normal folder object

responses tied to what part of the object is selected (icon or text) [Figures 8–9]. Different responses are also available for selecting of the folder or desktop backgrounds.

The current list of objects supported includes:

Folder Object : this is the standard folder which matches the concept of a physical folder on a real desktop. It can hold any type of object within it, including other folder objects. It can be either opened or closed. As mentioned, each object has different options for different mouse strokes as is shown in the drop down menu capture of a closed folder object [Figure 10].

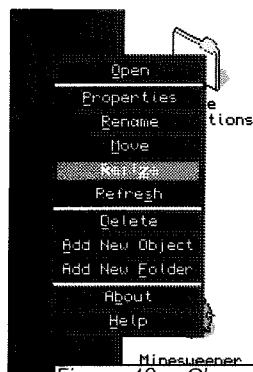


Figure 10 : Closed folder object drop down menu

Executable Folder Object : many programs are kept within directories or folders, containing other items such as databases, configuration files, etc. If stored in a normal folder, the user would have to open the folder to get to the executable

program. The Executable Folder Object looks like a normal folder object but with one additional characteristic. It includes a pointer to an executable object within the folder. The desktop behavior will treat it just like an program object but also allows it to be opened as a regular folder when necessary. This speeds access to the program, hides the complexity from the average user, but allows full access of the other contents when needed.

Program Object : This object points to an executable program which can be run directly from the desktop Program Object. It can be any executable binary or executable basic file.



Figure 11 : Program object drop down menu

The drop down menu shown here [Figure 11] shows the menu options for this object. The same program can have different objects, allowing for different command line strings, different data directories, or other configuration options. This is similar to being able to run the Files thing under the QPtr environment with different sort options or pointing to different drives.

Text and Graphic File Objects : any non executable file can have an object set to it which will open the file either with the default program for that type of object or a different specific program. QDT allows separate viewers to be set for both graphic and text files.

Unknown File Objects : these are similar to Text and Graphic File Objects except they do not have a default viewer or program to run them (simply because they are unknown types). The individual objects can be set to use the File Association defined program, a preset program at the time the object is created, or prompt the user for the program to use each time the object is opened/run.

Printer Object : this object is the result of user feedback from the North American QL show this past summer. This concept is still being developed. The basic idea is to allow any file to be dropped onto a printer object (for a specific printer) and have that file printed. The printer object will attempt to open a program to generate the printer output and then close it when done. If ProWesS is installed (which will be part of the installer options since it is now available at no

cost to everyone), I am thinking of the possibilities to pipe the results through its printing capabilities, taking advantage of its printer drivers. This should be possible for text files and perhaps some graphics files.

QDT Pieces – Things

QDT has several separate programs that do different things and are shared by all of its components. The most commonly used one is the DDMenu (drop down menu). This thing is set up to dynamically generate menus of different lengths and with different contents. The calling program sets all this up including the available and not-available entries. Here

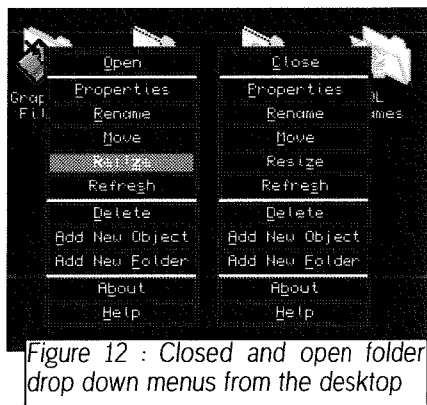


Figure 12 : Closed and open folder drop down menus from the desktop

are shown two drop down menus from separate folder objects on the desktop [Figure 12]. Notice that the closed folder starts with the 'Open' option while the open folder starts the menu with the 'Close' option. You can also see that the resize option is blocked out (not-available) for the closed folder since it the closed folder has no size to change. Two other common things are the About Window [Figure 13] which allows the calling program to fill in the details and the Error/Warning Message window [Figure 14]. The Error/Warning Message window also allows the calling program to set it as an Error or Warning, to fill in the details, and to return to the program or close everything down after being acknowledged.

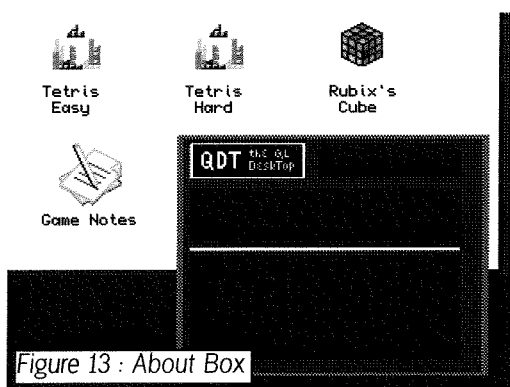


Figure 13 : About Box

Two other common things are the About Window [Figure 13] which allows the calling program to fill in the details and the Error/Warning Message window [Figure 14]. The Error/Warning Message window also allows the calling program to set it as an Error or Warning, to fill in the details, and to return to the program or close everything down after being acknowledged.

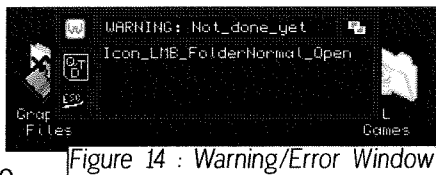


Figure 14 : Warning/Error Window

Design Notes

This effort has been and continues to be a challenging and enjoyable exercise in program and system design. While SMSQ/E (the current version of QDOS) has a lot of very nice features, it is also missing some basic capabilities. Thanks to many programmers, a lot of these have been compensated for and enhanced already. For a system design architect like myself, doing such a project is pure happiness (plus frustration, exhaustion, excitement, confusion, impatience... OK, you should have the picture by now). Trying to figure out how to get around the shortcomings and then finding some unique capabilities already build in is providing a constant challenge and reward environment. There have been a lot of design decisions and tradeoffs. Here are couple of the major design points that I have been following:

- 1) use everything possible that is already done (there is only one of me after all)
- 2) keep the user experience simple and positive
- 3) make it useable in a wide variety of QL systems but don't let backwards compatibility cripple any important feature
- 4) follow the open architecture concept (fully documented make useable by other programs)
 - this is tough to follow due to the time limits on the program; I want to ship it one of these days
 - designed with this in mind and possible to go back and make it fully open at a future date

As you have seen so far, I am heavily leveraging the QL's multi-tasking, the QPtr windowing environment and new features of SMSQ/E. File association and viewers now available will be used while I am considering requiring QMenu and QPAC for the final product. At this point ProWesS is not being required (except possibly for the printer object concept) due to system performance requirements. I felt that if I supported it I would also have to have a non-ProWesS version and there just isn't time to do both. ProWesS will run in parallel though with no problems.

Progress Report and Web Sight

QDT has been a very long and drawn out project. My real job (I am a senior hardware design engineer for Apple Computer) takes a tremendous amount of my time, along with my other interests and responsibilities. And then there have been

JOSCHEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
<http://www.j-m-s.com/smsq/index.htm>

QPC2 Version 3

Marcel did it again: major improvements to QPC! A long list of new features, which we won't list here, so please ...

see "News" in this issue!

To avoid confusion: If you already own QPC2 (Version 1), then the upgrade is NOT free! The naming is a bit confusing (we should have called it QPC3, but now it is too late, isn't it?), so here is the complete price list:

Upgrade to QPC2 Version 3

From QPC1 (or just QPC - the first QPC!)	£45
From QPC2 Version 1 (or just QPC2)	£29
From QPC2 Version 2	£13
(if bought/upgraded after 15. Oct. 2001)	£4

You will get a new manual (the QPC part)! Please send in your master disk for an upgrade. Please note: QPC1 will not be supported on this disk anymore! If you want to continue to use QPC1, please make yourself a backup copy of the master disk (you should do this anyway).

QPC2 Version 3 Full version £83

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99).

[Europe] £4,50 (if total value of goods is under £15 then only £3).

[Overseas] between US\$7.50 (1 item) and US\$17.50 (maximum).

All prices incl. 16% V.A.T. (can be deducted for orders from non-EU-countries).

Cheques in DM, EURO, International Money Order in DM or EURO (they all have to be drawn on German banks, Eurocheques and Credit Cards accepted. Please note that Eurocheques will not be valid anymore in 2002!

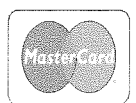
Prices are based on an average exchange rate of £1 ranging between DM 3,10 and DM 3,20.

Prices may be adjusted in case the exchange rate falls out of this range - in both directions!

We can charge your credit cards in £'s, US\$, EURO or in DM - please state the currency you prefer.

Please do not send any UK bank cheques in £ - our banks have increased the fee for handling them by 600% (no joke!) so we cannot accept them anymore, unless you add £6 for clearing the cheque.

E&OE.



the times where I got stuck with a piece of code that just kept breaking. Or the time when my cats dumped a glass of water on my PC laptop that I am doing most of this work on – multiple backups are now my mantra in life! The good news is that the progress has been steady and all the images you see in this article are captured screen dumps generated by real QDT code. All the pieces that have captures have a good share of their code completed. The two exceptions are the Configuration Notebook and the Desktop Builder (actually cheated and manually entered the main window list of objects in the Desktop Builder capture – but the icons and structure are all real). The configuration notebook still requires the page designs and eventually the config block and external config file development. The Desktop Builder requires primarily the parser to be moved from QDT to it and a bunch of linking. The pieces that don't have pictures for the most part have not been started. The core design work has been completed for all of them except the printer object. In other words, stay tuned. At this point things are coming together nicely (it is great finally seeing real results on the screen and under the mouse!) and I will be having updates on a regular basis. To help keep interested parties informed, I am in the process of generating a websight. I have established a domain name (www.jdh-stech.com) and will have a QDT set of pages linked to from this location. There will also be an email link from that sight set up exclusively for QDT questions/suggestions.

Help Anyone?

There are a few things that I could use the assistance of people who understand them (coding examples are great if available). These are things that I can figure out but it will take additional time to do so. If someone can assist on any of them, it just means that I can spend more time on the core QDT code. Here is the list of items that I need help with.

- Things in 'C'
- New ptr calls for wait for return event in 'C'
- Full color in pointer windows in 'C' (or assembly if not available)

The Full Color item may need additional explanation. At this point, I can only get the code to use the original QL color scheme for the basic qPtr window pieces instead of the really cool 256 colors that I use in the custom parts of the desktop. It would really enhance the look and feel of the desktop if all colors could be put to use on all parts of the windows. If this isn't possi-

ble, it is likely that I will eventually implement totally custom windows to allow for the desired color and shadow effects. When I get more of the pieces put together, I will be calling for a very limited number of Beta testers. I will post the request on my web site and through the Sinclair usenet group.

System Requirements and Release Plans

For any capabilities this advanced, there will be minimum system requirements. Here are the QDT hardware and software requirements that I am anticipating. Please note that the minimum requirements are just that – minimum. Many of them will impact the quality of the user's experience with QDT but at least it will run.

Hardware [optimal - minimal]:

Hard Drive – Flash Rom Drive
(floppies will probably be too inconvenient)
Memory – unknown at this time
Resolution: 640x480 or higher – QL standard res.
Color Mode: Hi Color – QL 4 Color Mode

Software:

Operation System: SMSQ/E
Windowing System: Pointer Environment
Included Software Packages: QMenu, QPac (?)
Freeware/Shareware: selected viewers, file association

The initial release will be built to only support hard drives and will likely have some issues with the QL minimum resolutions. The project is designed to eventually support all of the above listed requirements. However, to get it out the door, I am handling and doing initial optimizations for hard drives and higher resolution only. A second release (hopefully not too far behind the first release) will add the Flash ROM and QL standard resolution support. The QDT design is being done with these 'low end' configs in mind. In addition, it is expected that the first release will have all the basic functionality but without all the nice options, such as the Tab Launchpad. For those who purchase QDT, the extra pieces and feature enhancements will be available through the internet at no charge or through dealer upgrades. I am expecting the enhancement period to last for around 6 months to a year after the launch of QDT.

Closing

To close this off, I would like to give credit where it is due. First of all, if it weren't for all the people

who have stuck with and kept enhancing the QL's capabilities, none of this would be possible nor would I have been interested in attempting this project. To point a final finger of blame, you can all thank Jochen Merz for my actually working on this project. In a fit in pure insanity, I told Jochen years ago that if the QL ever got real hi color drivers, I would write this desktop. Jochen was very happy to send me an email to let me

know that the drivers were soon going to be available and to ask how soon QDT would be done. If all goes well, QDT should be available in time for next summer's North American QL show. The original intent was for this holiday season, but the fates (and human limitations) ruled against that. Again, please watch the new sight and feel free to pass on any suggestions that you may have.

Gee Graphics! (on the QL?) - Part 25

HL Schaaf

Back to Graspix

There must have been some changes to SMSQ/E since the Graspix_bas exercise (GG#3, March/April 1998). It shows up in the MODE 8 testing when BLOCKS and PIXELS don't always match up any more. This also shows up in the latest version of QPC2.

After doing some noodling around I found that the newer versions of SMSQ/E allow the entry of odd size BLOCKS in odd locations. In the earlier versions the QL must have had an internal checking system that automatically converted any odd inputs into even values by subtracting 1.

To demonstrate the effects, try the listing OddBlocks_bas. It will show two BLOCKS and in the window below will give the width and column and ink colors for each, one above the other. See if they match your expectations.

If you have access to later versions of SMSQ/E or use QPC II v2.03 you should see some interesting fringe effects along the left and right edges of the BLOCKS.

To keep Graspix_bas viable merge the listing Graspix_bas_later8 into the original Graspix_bas from the cover disk that came with that issue. Then you can check that it does work properly.

Now there are many more setups I haven't tried such as Aurora, QPC in high colors and extended resolutions, Q40, etc. I'd be grateful if those who have such systems would take a quick look at OddBlocks_bas and tell us the results. It may take a good magnifying glass to see the fine detail. And I'm sure that getting Graspix_bas to work on the high color and/or resolution systems other than 512 by 256 will require extensive revision. Maybe we can start on that for next time, or get back to the matrix routines.

Listing OddBlocks_bas

```
100 REMark OddBlocks_bas
110 REMark HL Schaaf 5 Nov 2001
120 REMark for GG#25
130 REMark to exercise BLOCK command
140 REMark with operating systems
150 WTV : MODE 8 : PAPER 0
160 FOR wid = 1 TO 5
170 FOR col = 24,25
180 FOR nk = 1 TO 7
190 CLS : CLS #0
200 PRINT #0;'width = ';wid!!'column =
    ';col!!'ink = ';nk
210 BLOCK wid,25,col,25,nk
220 PAUSE 10
230 BLOCK wid-1,25,col-1,50,nk
240 PRINT #0;'width = ';wid-1!!'column =
    ';col-1!!'ink = ';nk
250 PRINT #0\\"touch [spacebar]"
260 PAUSE
270 END FOR nk
280 END FOR col
290 END FOR wid
300 CLS#0
310 REMark end of listing for OddBlocks_bas
```

Listing Graspix_bas_later8

```
100 REMark Graspix_bas_later8
112 REMark revised October 30, 2001
115 REMark correction for later versions of
    SMSQ/E in MODE 8
212 later8 = 0
215 IF (VER$ = "HBA" ) THEN
216 IF (VER$(1) > 2.89): later8 = 8
217 END IF
2275 IF (Mo_de = later8) AND (blkcol% MOD
    2) THEN
2276 blkcol% = blkcol% -1
2277 END IF
2535 IF (Mo_de = later8) THEN
2536 GraphicX = (Block_Column%+.25)*H_rpp
    + Left
2537 END IF
2780 REMark end of listing for
    Graspix_bas_later8
```

Two Users

Geoff Wicks

Two issues ago Wolfgang Lernerz put me in the firing line because I had "imprudently" voiced "the opinion that speed is of no essence (as long as you have reached a sufficient speed, e.g. Gold Card), and that today's machines have more than enough memory etc."

Wolfgang then went on to describe his own system, obviously QPC2 or a Q40, which has 32 Mb with only about 20 Mb free. He uses it in a busy office in which he is constantly interrupted. He wants his programs to be available quickly and easily, and makes extensive use of the QL's task sharing capabilities. Even when not developing programs, for which he needs compilers and editors, he daily uses fax, contract management, case management, accountancy, calculator, word processor, spell checker and file management programs. To do this he always loads 35 extension files including menu extensions, QMON, QPAC2, fileinfo, Qlib, Prowess and the Data Design engine.

The first thing I did after reading Wolfgang's comments was to start work on transferring Quill files for a customer who was having difficulty in using QPC and QL-2-PC Transfer. Until recently his system had been a black box and Trump Card, and practically the only program he uses is Quill. He had typed tens of thousands of words into Quill describing his life and experiences, and his family, anxious to preserve these documents, had persuaded him to upgrade to a PC. Although both another trader and I had had extensive contact with him, he had still been unable to transfer his work.

Throughout my contact with this customer, his name had seemed strangely familiar. I assumed I had heard of him via QUANTA or possibly QL Today, but when I started to transfer his material, I realised it was not through a QL connection. This man had risen to the top of his profession, and I had probably first heard of him about 10 years before the release of the QL when Sinclair Research was a relatively unknown company producing the black watch.

You could hardly have two QL users further apart in their equipment, software and daily use of the QL. At the same time you could hardly have two QL users more alike. Both are intelligent, capable people who are loyal QL-ers with little desire to use another machine. Most of us lie between these two extremes. We want neither the complexity of Wolfgang's system nor the simplicity of my clients. We want a system that serves our computer needs as simply and as cheaply as possible. Indeed, Wolfgang was misquoting what I had written. I did not write that speed was not important, but that "for most QL users speed ceased to be a major issue with the arrival of the Gold Card".

Now let's look at a few statistics. I estimate that fewer than one QL reader in five and fewer than one QUANTA member in five has bought a Just Words! program. Probably well under one QL user in twenty has ever used a Just Words! program.

This is not a moan that I have been let down by the QL community. It is the way it should be. Just Words! produces specialist software, which by definition will have a minority appeal. It has succeeded in its aim of having a small, but enthu-

siastic, group of users. Other QL users sometimes apologise for not buying my programs, but why should they? It is not disloyal to either me or the QL community if you do not buy something you neither want nor will use.

In practice the QL community is a network of minority groups. If you want a challenge, you could try listing and classifying these groups. You start easily with two broad categories, the hardware enthusiasts and the software enthusiasts, but from then on it gets complicated. There are hardware developers and hardware tinkerers, computer builders, repairers and peripheral specialists. On the software side you have program writers and routine writers. There are superbasic, machine code, C and Linux specialists. There are devotees of SMSQ, QDOS, QDOS classic and Minerva, and lovers and haters of the pointer environment. There are also a lot of people, maybe the majority group, who just use the QL as a computer to further their hobby or other outside interest.

Somehow it all hangs together, although often these different groups have little contact with one another. I suspect some even see themselves as being higher in the "pecking order" than others, and am amused by the attitudes of some experts to the QL email users group. One does not subscribe because it is too "twee" and another because it is "too superficial", yet most of us understand the more technical hardware and programming topics? How much do the "experts" know of or understand what goes on outside their specialist field?

The big danger for QL users, and especially the experts, is in overexaggerating the importance of our own contribution

to the QL. It may be valuable and vital to the future of the QL, but it is not necessarily the first priority for the average QL user. Some of the fierce debates that take place over the internet or in QL publications are important for improving and maintaining standards. But I suspect most are only of interest to the protagonists and leave the wider QL community unmoved.

My client with the Trump Card and black box is just as important to the survival of the QL as any of the experts. His problems were not entirely of his own making. When I started to transfer his work, I too found it impossible initially, and it was sometime before I realised the cause of the problem. He had formatted HD disks on the Trump Card, which treats them as DD disks, and this confused QPC. To add to the compli-

cations his version of QL-2-PC Transfer contained a bug that prevented the transfer of large Quill files, and he had missed out on the free upgrade that corrected that bug.

More importantly, that user, after years of using a Trump Card and black box, had upgraded his system. Indeed, Roy Wood and I know of other people who have done just that in order to transfer their files to a PC. Provide a product that people want and they will not only buy it, but, if necessary, will upgrade their system.

I do not intend to be disrespectful to the hardware developers, but the average QL user is probably more concerned about a reasonably priced QL compatible printer than the latest, fastest and biggest memory QL product. With software, unfortunately, it is the other way round. Many

would like more and better graphics, word processing and internet programs, but little software is currently being produced, and most is relatively simple.

Those of us who produce QL hardware and software, or who write regularly for QL Today should always remember we are in the minority. As any trader can tell you there are many QL-ers out there who have different needs, ideas and priorities from those of the experts.

By the way, Wolfgang Lenerz was wrong when he accused me of writing "imprudently" for QL Today. I think carefully about what I write and never write imprudently. If however, Wolfgang had accused me of writing "impudently", he would have been spot on. I often write impudently for QL Today.

A new QL Publication

Georgio Garabello

Sadly, we can't say that there are many publications related to the Sinclair QL personal computer, but also in this little world, from time to time, some good news comes up.

Here we wish to talk about the birth of a new electronic magazine, dedicated to the QL platform. Its name is QL Magazine, and it's written in Italian language; as I said before, it's an electronic magazine, which can be read from your computer. It's written in Italian, because its writers are Italian and the magazine itself is dedicated to the Italian QL users.

THE BIRTH

QL Magazine was born officially on the Eight of October 2000, when we presented it at the 9th Italian Sinclair QL meeting. The magazine comes out every three months, with a fixed number of ten thematic articles, plus some periodical surveys.

OBJECTIVES

QL Magazine tries to support the Italian (and

other) QL users by diffusion of news, information, programs and technical notes. The magazine doesn't only offer news about the latest products or events regarding the Sinclair QL, but in every issue are included one or more free application packages, images, and an announcement page. Inside every issue of QL Magazine there is a section dedicated to the ZX Spectrum, because many QL users are also devoted to its elder brother. Choosing the Italian language for the magazine content was due to practical considerations, because of the lack of collaborators and the little time for writing. Indeed, all the contributors, work at the magazine in their (little) spare time.

THE MAGAZINE ITSELF

QL magazine can be read from any QL platform. It can be run from the FLP1_ device, inserting a disk containing the magazine before turning on the QL, or directly via the command LRUN FLP1_BOOT.

Obviously, it is possible to run the magazine from other devices (for example, from a subdirectory of your hard disk), by giving the correct value to the DEV\$ variable, placed at the very beginning of the BOOT program.

Every issue is made up with a summary menu, reporting the table of contents of every article, image, and contributions inside the issue, all in different colours to better distinguish an argument (and give some more "colour" to the output). In the index page there can be more articles than a screen page can display: it's possible to scroll the page by pressing the up and down arrow keys.

If the list is long, it's possible to move from page to page by pressing the Shift key before tapping Arrow up or Arrow down keys. To select an article, or an image, or even a program, just press the Enter or Space keys (or the relative mouse or joystick buttons).

It's possible that, if you try to launch a program, this won't work correctly. This is not an error, but depends on the program itself: it may need some system add-on you haven't installed or not supported by your machine (i.e. the called program may need the Pointer Environment: if this extension is not loaded, obviously the program will abort running).

An important note: before launching a program, you **MUST** know how to exit it: at first sight you might find this advice a nuisance, but keep in mind that executing an application you don't know might be dangerous to the system. Be careful!

The image viewer used by QL magazine is Photon: to exit the viewer press the Enter key. For more details about using the program, please consult the instructions contained in the magazine.

It is possible to stop reading articles by pressing the Esc key: same applies to exiting the magazine issue. Read the instructions of any program you want to try, to know how to use/exit them. Some of the articles in the magazine are repeated in all the issues (like the instructions on how to use the magazine and the copyright information file): should some of these be modified, the word "Aggiornato" will appear next to the article title in the index page.

MULTI-PLATFORM

QL Magazine is designed to be multi-platform. Every issue provides an interactive viewer to be used with any Sinclair QL, but it can be read with any other machine and operating system (such as Microsoft Windows, Apple Macintosh, Linux, Amiga, Risc PC, IBM OS/2) if you have got a simple HTML browser: the magazine has got an HTML index page in which are listed articles, images and programs. An online version of QL Magazine, though planned at its birth, is for now suspended, due to the actual Italian laws concerning publishing trade. Should it be possible, we will as soon as possible publish any issue on the World Wide Web.

HOW TO OBTAIN THE MAGAZINE ISSUES

You can freely download every issue of QL Magazine from the following Internet address:
<http://utenti.tripod.it/Sinclair/qlm>

Using INdEXfindString with DATAdesign

Wolfgang Lenerz

A few years ago, when visiting a bookshop and using their computer to search for predetermined book, I came upon a feature that really impressed me. Indeed, as soon as I input a single letter in the search field, the computer, (a PC) would propose an author starting with that letter. Add another letter, and an authors name starting with these letters would be proposed. This happened while I actually was typing the authors name. Even if I don't type quickly, the succession of proposed names did come blindingly fast, in the time it took to type one letter..

This was quite neat, because, once you had typed three or four letters (thus refining the search), it generally proposed the author you

were looking for, and you just had to accept that with enter.

I thought "wow, I want that on my QL, too." I always use Datadesign for my database work, so I wanted to do that there. As it happens, I have a database on books I own (it is, admittedly, a very small database, with about 1000 entries) and I thought it would be nice to have that kind of possibility there.

But, try as I might, it just wasn't possible at that time. Datadesign had two ways of finding entries: either a very fast way, i.e. INDEXfind, or (as the manual says) a slow,linear but flexible way, i.e. the FINDxxx commands. Both ways were unusable for me: FINDString is simply too slow - it will search the entire database from the current record each time, and if the name searched for is the last record, it'll just take too much time, even on a small database like mine.

INDEXfind, of course is MUCH faster, but, as I discovered, unfortunately too inflexible:

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost..... **£24**

*****NEW*****

superHermes

A major hardware upgrade for the QL

All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) **£90 (£92/£93)**
 IBM AT UK layout Keyboard..... **£11 (£13/£15)**
 Serial mouse **£8 (£8.50/£9)**
 Capslock/scrolllock LED **£1 (£1.50/£1.50)**
 Keyboard or mouse lead **£3 (£3.50/£3.50)**
 High speed serial (ser3) lead **£4 (£4.50/£4.50)**

Hermes available for **£25 (£26/£27)** Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead) **£53 (£54/£55)**

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send **£3 (+£5 for manual if reqd).**

Send disk plus SAE or two IRCs

MK1...£40 (£41/£43) MKII...£65 (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39 (£40/£41)**
 4mbytes RomDisq..... **£65 (£66/£67)**
 8 mbytes RomDisq..... **£98 (£99/£100)**
 Aurora adaptor..... **£3 (£3.50/£4)**

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost **£34 (£35/£36)**

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) **£40 (£43/£44)**

4 amp total (for motors etc) **£45 (£48/£50)**

Relays (8 3a 12v 2-way mains relays (needs 2a power driver) **£25 (£28/£29)**

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25 (£27/£28)**

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting **£30 (£31/£32)**

Temp probe (-40°C to +125°C) **£10 (£10.50/£11)**

Connector for four temp probes **£10 (£10.50/£11)**

Data sheets **£2 (£2.50/£3)**

Control software & manual (for all I/F) **£2 (£2.50/£3)**

QL SPARES

Keyboard membrane **no longer on sale**
 1377 PAL **£3 (£3.50/£4)**
 Circuit diagrams **£3 (£3.50/£4)**
 68008 cpu or 8049 IPC **£8 (£8.50/£9)**
 8301/8302 or JM ROM or serial lead **£10 (£10.50/£11)**
 Power supply (sea mail overseas) **£12 (£17/£21)**

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe/Rest of world). Payment by cheque drawn on bank with UK address/ postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details

10 May 01

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: 01442-828254

Fax/BBS: 01442-828255

tony@firshman.demon.co.uk

http://www.firshman.demon.co.uk

INDEXfind requires that the entire data for an index be given. In the case of a character field, such as the one I am using for book or author names, this MUST consist of the first 8 characters (bytes) of the name. Indeed, for string (and long) fields, INDEXfind uses an 8 bytes index – and the entry searched for must match the 8 bytes. Clearly this couldn't work, since, when the user enters the first character, I only have one character to search for.

After some headscratching, I decided to write a new command for Datadesign. I then sent it to Joachim Van Der Auwera, who very graciously allowed this to be merged into the Database engine. so that later versions of Datadesign now have the "IndexFindString" command (thing name: IFDS). This command allows you to pass only part of a name and an indexID, and it will return the recordID of the first record where the first letter(s) of the field used for the index correspond to the letter(s) used as search criteria. As it uses the index, it is pretty fast, too – sufficiently for what I had in mind.

The (Basic) syntax of the command is simple:
recordID = IndexFindString(#bufferID,string\$)

It uses the default index of that buffer ID. You can also pass it an indexID instead. The string\$ parameter are the first (1 to 8) characters of the string to be found.

Of course, this command will only work for Text fields (and you don't need it for other fields). Also, it can only find records where the first characters match the search string – if you gave 'wo' as search string, it would find 'wolfgang', but not "roy wood".

Once this command was finished and working, I then wrote a small basic procedure to help me with it. After all, it isn't enough to find the records quickly, the user also has to type in the names, see the proposal and be able to accept it with Enter.

The general theme I adopted thus is this: I always use the pointer environment. The data is shown in loose items. The user can click on the field/loose item that interests him. The item is blanked, a cursor appears and the user can start to type. As soon as a character was typed in, a name is found and printed in the item. If the user accepts that with Enter, all is ok, else he can continue typing, a new name will be proposed etc. The user cannot edit the text (except for backspace) not move around the text with the cursor keys.

OK, so much for the general description. Here is the basic procedure. I've line-numbered it for easy reference, even though, using the Basic Linker, I generally don't use line numbers.

```

100 DEFine PROCedure find_record (which2%)
110 rem try to find entry in database by proposing items
120 LOCAL a$,temp%,lp%,chan%,tot$,found$,mlen%,lp2%,flag%,which%,recid
130 local my_index,I_have_no_index%
140   I_have_no_index%=1           : rem no index used yet
150   select on which2%
160     =author_it%:my_index=ii_main:I_have_no_index%=0
170     =title_it%:my_index=ii_titles:I_have_no_index%=0
180   end select
190   DEFindex#bi_main,my_index
200   FIRSTrec#my_index           : rem start on first name
210   chan%=FOPEN('con')         : rem open channel...
220   DR_LWDF#chan%,main_defn,which2% : rem ...over item
230   PAPER#chan%,menu_pap
240   INK#chan%,menu_ink
250   CLS#chan%
260   CURSEN#chan%               : rem make sure there is a cursor
270   a$="":tot$="":mlen%=0
280   which%=which2%-author_it%+1 : rem the field to search!
290   REPEAT lp%
300     a$=INKEY$(#chan%,-1)     : rem get one char
310     temp%=CODE(a$)
320     flag%=1
330     SElect ON temp%          : rem now let's see what we do
340       =0,10,27:EXIT lp%      : rem end
350       =32 TO 191             : rem 0-9,a-z,A-Z & accented chars
360       tot$=tot$&a$           : rem new total word to find
370       mlen%=mlen%+1         : rem new length of string
380       IF mlen%>8 or I_have_no_index%
390         PREVrec#my_index     : rem previous record, since we search with "a"
400         REPEAT lp2%
410           found$=tot$
420           FINDstring #bi_main,which%,"a",tot$ : rem find it

```

```

430         IF dd_err:flag%=0:EXIT lp2%
440         found$=GETline$(#bi_main,which%,1) : rem get data from database
450         IF found$(1 TO mlen%)==tot$:EXIT lp2%
460     END REPeat lp2%
470 ELSE
480     found$=tot$
490     recid=IndexFINDstring(#my_index,tot$)
500     IF dd_err<>-10
510         GETrec#bi_main,recid
520         found$=GETline$(#bi_main,which%,1)
530     ELSE
540         flag%=0
550     ENDIF
560 ENDIF
570 IF not flag%:empty_all
580 CLS#chan%
590 PRINT#chan%,found$; : rem print name now
600 CURSOR#chan%,mlen%*6,0 : rem set cursor to chars we have (name might be longer)
610 IF flag%:get_record 1 : rem fill record
620 main_lfl%(which2%)=0
630 DR_LDRW main_defn,main_lfl% : rem redraw these
640 =194 : rem backspace
650 IF mlen%
660     IF mlen%=1
670         tot$=""
680         mlen%=0 : rem there is no search string
690         empty_all
700         cls#chan%
710         cursor#chan%,0,0
720         main_lfl%(which2%)=0
730         DR_LDRW main_defn,main_lfl%: rem redraw these
740         next lp%
750     ELSE
760         mlen%=mlen%-1 : rem new searchstring
770         tot$=tot$(1 TO mlen%)
780     ENDIF
790 IF mlen%>8 or i_have_no_index%
800     FIRSTrec#my_index
810     REPeat lp2%
820         found$=tot$
830         FINDstring #bi_main,which%,"a",tot$ : rem find it
840         IF dd_err:flag%=0:EXIT lp2%
850         found$=GETline$(#bi_main,which%,1) : rem get data from database
860         IF found$(1 TO mlen%)==tot$:EXIT lp2%
870     END REPeat lp2%
880 ELSE
890     found$=tot$
900     recid=IndexFINDstring(#my_index,tot$)
910     IF dd_err<>-10
920         GETrec#bi_main,recid
930         found$=GETline$(#bi_main,which%,1)
940     ELSE
950         flag%=0
960     ENDIF
970 ENDIF
980 IF not flag%:empty_all
990 CLS#chan%
1000 PRINT#chan%,found$; : rem print name now
1010 CURSOR#chan%,len(tot$)*6,0 : rem set cursor to chars we have (name might be longer)
1020 IF flag%:get_record 1 : rem fill record
1030 main_lfl%(which2%)=0
1040 DR_LDRW main_defn,main_lfl% : rem redraw these
1050 ENDIF
1060 END SElect
1070 END REPeat lp%
1080 CURDIS#chan%
1090 CLOSE#chan%
1100 DR_LDRW main_defn,main_lfl% : rem redraw these
1110 END DEFine find_record
1120 :

```

The philosophy of the procedure is that you can use it with several indexes, so that you can index more than just the author names. In my case, I also index the book names. Hence, you pass a parameter „which2%“ to the procedure. This parameter correspond to the loose item number. Thus, after setting a flag in line 140 (more of which later), I select which index to use, depending on the parameter passed, i.e. the loose item clicked, and then define the index selected as default index for the database. The bufferID of the database is held in a global variable called "bi_main", and I make sure that the search will start from the first record in that index (line 200).

I then open a window over the loose item the user clocked on, and set the paper and ink colours using global variables menu_pap% and menu_ink%, which I also use when defining the PE working definition of the window. Using global variables like that makes it easy to implement the traditional PE colour schemes, you just change the values of the variables to fit the colour scheme. I also make sure that there is a blinking cursor in the window (line 260). After initialising some variables, I determine which field in the database corresponds to the item clicked (line 280). This should normally have been done in the select clause in lines 150 to 180, but it so happened that a simple calculation was enough. Then comes the loop. It waits for the user to input one character, converts that character into a number (line 210) and uses that number in the select to determine what to do. There are three groups of characters input by the user: those that mark an end to the user's interrogation (Enter=10, ESC=27 and I also used 0 for good measure), those that correspond to valid characters (32 to 191 – this also allows accented characters) and the backspace key (194, in line 640). If the user marks the end of the interrogation, we quit the loop. More of that later. If the user enters a normal character, we first of all compose the total word for all characters entered so far, and set the length of that word (line 370).

Then a test is made (in line 380, more of which later). Let's suppose, for the time being, that the test directs us to lines 480 and following, which would be the case after the user entered the first (few) characters in the window. In line 490, I use the IndexFindString function to search for the corresponding record. I check whether a record was found (line 500) and get the data of the field (i.e. the author or book name) into the variable found\$ (line 520). If an error occurred, I clear all the windows showing data (line 570, this calls a

procedure "empty_all" which I haven't included, but that just does as I said, i.e. clears all the windows and some variables).

I then clear the window and print the name out in it. Next, I set the cursor in that window to point after the character the user has just entered, so that he isn't confused (line 600). Then I fill in some other loose items with the data from this record (line 610), reset the item the user clicked on to available (line 620) and redraw all items (to show the ones I also changed).

That's the heart of this procedure. To come back to the test on line 380: This tests two things: first, whether the name we're searching for is longer than 8 characters (mlen%>8) and second whether we use an indexID at all (the variable l_have_no_index% is 1 if no index ID is used, else 0 – see lines 140 and 160,170).

Let's take these in this order:

If the name we're looking for is longer than 8 characters, then IndexFindString, and IndexFind, are useless – because the search criteria used for the indexes in Datadesign is never longer than 8 bytes. If your field contains, for example first names, then searching for 'wolfgang' (8 chars!) or 'wolfgangxxx' with IndexFind or IndexFindString will give you the same record! Clearly, the is not what the user wants here, if he enters a name longer than 8 chars, he wasn't satisfied with the names proposed up to now. So how do you solve that problem? Well, you use the slow Findstring command – which is what I wanted to avoid in the first place, so why am I using it here? Because now it will be faster to find a matching record. FindString works from the current record onward. The current record already matched the first 8 characters of the search string, so the next one matching 9 chars won't be far away... You have to make sure, though, that, in the record found by Findstring, the string found matched exactly at the beginning (line 450) else you keep searching.

The second test condition was whether we had an index. Indeed, this procedure allows you to search those fields that do have an index, and those that don't. If an index exists for the field corresponding to the item clicked, then that index is selected (lines 150 to 180) and used with IndexFindString. If no index exists, you can still search, but only using the slower Findstring. That way, the same procedure can be used for all searches.

Now we come to line 640 and following, treating the case of the backspace key. We don't do anything if the backspace key was used when

no name was type in yet at all (line 650). If a name was already typed in, and the user wants to amend it, we first of all check whether the character deleted was the first (and thus only) character of the name, in which case no search criteria exists any more, the window is simply cleared and redrawn (lines 660 to 730) and we return to the loop (line 740, early return). Else, we make the new search criteria and use the same code as earlier on to find the corresponding record (lines 790 to 1040 are practically the same

as lines 380 to 640). I have duplicated the code to speed thing up a little, I could have put the duplicated code in another procedure and called that procedure from both places.

At the end of the loop, to which we come when the user hits enter or ESC (line 340), we disable the cursor, close the channel and redraw the items for good measure.

I hope you can find this code useful in your own projects!

QL Forever W. Lenerz

German/Austrian QL Show in Berchtesgaden

Jochen Merz

Early October, a fairly large number of QL users from Italy, England, Austria, Switzerland, The Netherlands and Germany turned the German/Austrian QL Show 2001 into a success. The venue was a "Gasthof" right in the centre of Berchtesgaden, surrounded by a breath-

taking view in every direction. Many thanks to Friedemann Oertel for organising the show and for booking such excellent weather!

I spent the week before the show with my Girlfriend in Schönau am Königssee (next to Berchtesgaden).

We did some sightseeing, and it was quite difficult to choose what to do because Friedemann has provided the visitors with a long list of famous and nice things to visit. We could not squeeze everything into this week, so we know we'll be back - probably and hopefully next year!

Berchtesgaden is more or less surrounded by the Austrian border, so it was obvious for the English dealers to fly to Salzburg Airport instead of Munich.

QBOX-USA BBS



Operating since 1993 on a Sinclair QL from Utica, Michigan, USA

Supporting ALL Sinclair and Timex users

Message and File Areas for QL, Z88, Spectrum, TS2068, ZX81, TS1000

Modem speeds 300 bps to 33.6k bps supported

24 hour operation - call anytime

810-254-9878

When they arrived at the airport, they gave me a call. I quickly unloaded the car (all my computing stuff and goods, up to the 3rd floor). Picked them up at Salzburg and we had a nice evening ... more and more QLers arrived Friday evening and the table was growing and growing.

Next morning, I had to load the car, drive to the venue, unload everything. After the show, everything had to go back to the car, up to the apartment again to make room for Tonys Roys and Bills stuff, 'cause they had to go back to Salzburg airport on Sunday. Then, everything down to the car again, later home, and unloading everything in a rush after the 9 hours drive. I think that's what gave my back the problem, because the next day I got a lumbago (quite serious) which, unfortunately, stopped me going to the Paris show. Took about two weeks to recover. I am very sorry I missed Paris, but there was no way I could sit in a car - not even for 10 minutes (could not leave home for the first 5 days).

Back to the show: it was very well attended, users from many countries came to the show. The whole of Saturday was used as the "main" day, and it started before 10 and did last until after 8pm, VERY long and unusual. Marcel Kilgus was there too, talking to QPC users and getting more ideas for the next version (which are implemented already, see NEWS!). The pictures give you an impression of the room at various times. The "main" part ended with the group having dinner together - nice, funny and interesting talk like we know it from the "southern" shows. Some users continued 'til very late, but the majority went to bed.



In front of the venue was a Biergarten, and the sunny and warm weather invited many users to have a break outside. The whole day, although it was very long, felt like it was only several hours - very busy and entertaining. Some of us formed a group and had a look at "Eagles nest"

on top of one of the mountains on Sunday morning. It was sunny and hot again, and the view was excellent. After that, most users went home. It would be nice to have another show in Berchtesgaden again. It was well organised, and everybody who considers coming again should

reserve a couple of additional days or turn it into a short holiday. Pity that US visitors had to cancel their visit because of the September attack - I am absolutely sure they would have enjoyed the meeting and the area as much as we did. Looking forward to the next show!

Error Trapping comments

Geoff Wicks writes:

In his otherwise excellent article on Error Trapping, a neglected subject in QL publications, David Denham has missed some of the QLiberator error trapping possibilities.

A program compiled with version 3 of QLiberator using WHEN ERROR can still be run on the early QL ROMS that do not have the keyword implemented.

Page 15.5 of the QLiberator manual:

"A program which uses WHEN ERROR can only be entered and compiled on a system with JS, MG or Minerva ROMs. However the object programs will run on any QL provided that the procedure REPORT is avoided. QLiberator will produce compatible code to support use of ERLIN, ERNUM and all the functions which test for specific errors such as ERR_NF even though those functions are not present in the AH and JM ROMs."

To JS, MG and Minerva ROMS we can, of course, add SMSQ.

George Gwilt writes:

I was very interested in David Denham's article in the last issue of QL Today (v6.3), especially in the section about Turbo. I must confess that so far I have only used one level of error trapping, so I thought I would try out David's "useless little test program".

The program (on Page 31) is supposed to react to errors of numerical input by first printing "Oops ..." from the WHEN ERROR 0 clause in lines 110 to 150. It should then proceed to the WHEN_ERROR 1 clause and print "...silly!", but I just could not get that to happen...

... until I dug into Turbo and made a tiny, almost invisible, alteration. In fact I altered a "bgt" branch to "beq". Miraculously the altered Turbo allowed David's program to work - almost.

I think it may well be confusing, but the fact is that when an error occurs, Turbo looks for the last WHEN ERROR 1 which occurred. It obeys the instructions inside that clause and, if all the instructions are obeyed up to the END WHEN, without any RETRY, or it obeys a CONTINUE, or if a further error occurs, the latest WHEN_ERROR 0 clause is invoked.

This means that David's program as it stands will ignore the first, WHEN_ERROR 0, clause and just ask for another go "...silly!".

A further point, that is not yet documented as far as I know, is that an END_WHEN automatically generates a CONTINUE, which therefore need not be put in a program if it would be the last instruction in a WHEN_ERROR clause.

With the 0 and 1 switched, the CONTINUEs eliminated and (a) replaced by (entry) in line 220 the program (listed below) works as intended if compiled by Turbo v4.11.

```
100 IF COMPILED
110 WHEN_ERROR 1
120   PRINT "\"Oops, due to error
      "&ERNUM%&" at line "&ERLIN%
130   PRINT " this program will now
      close down."
140 END_WHEN
150 WHEN_ERROR 0
160   PRINT"Enter a number, silly!"
170   RETRY
180 END_WHEN
190 END IF
200 CLS
210 FOR entry=1 TO 10
220   RETRY_HERE:INPUT"Enter number
      "&(entry)&" > ";number
230 END FOR entry
240 IF COMPILED
250   WHEN_ERROR 1
260   END_WHEN
270 END IF
280 CLS:SUSPEND_TASK 100:STOP
```

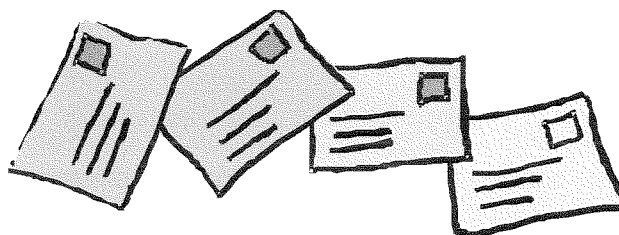
Other alterations I made were to replace ";" by "&" in line 120 and in most of 220. This makes for a

faster program. I added a "\n" to line 120 just for looks. I put the two WHEN_ERROR clauses inside one IF COMPILED clause. Finally I added a last line, 280, just to give a short cleared window

before it is whisked away by the Pointer Environment when the compiled program stops.

My thanks are due to David Denham for forcing me to discover the error in Turbo.

Letter-Box



Norman Dunbar writes:

QL Filename Extensions

Missing in action are:

- _exe Same as EXEC
- _pm2 Printer Master 2 print driver.
- _o Intermediate output from C68 compiler. Relocatable file ready to be linked by the linker.
- _s Intermediate output from C68. Contains C code converted to assembler, ready to be assembled by AS68.

I'm afraid I can't help with the ones listed as unknown in the article though.

The Ultimate Sin?

Geoff's article summed up the entire Sinclair 'thing' very well. I'm sure he'll have plenty of replies, but I recon he was just about spot on. The ZX80 was good, but buggy. The ZX81 – upon which I started my long and 'distinguished' career in computing – was great, and buggy. The Spectrum was probably the best all time home computer and sent thousands on their way to a new career I suspect, it too was buggy. The QL was buggy, overpriced and doomed to fail – the media had become fed up of Sir Clive by then and took almost every opportunity to trash him and his work – esp after the pub incident with the guy from Acorn.

Then there was the flat screen tv – which worked, but needed a lens over the screen to use, the black watch – which ate batteries when working and the matchbox radio, which I think worked! The calculator (Cambridge Scientific?) used reverse polish notation (like the Forth language) so '1+1=' became '1 1 +' which was quite difficult for many people. Plus, it was buggy! Divide zero by zero for an interesting display which counted up from 0 to its maximum number – I think it was 99,999,999. More entertaining than an error message though!

I suspect that the biggest part in his downfall was the C5. The blurb from Sinclair talked about

a new electric car, not a pram on wheels. The whole thing made him such a laughing stock in my opinion. It was never a car, and it was very dangerous to use on any sort of busy road.

Apart from all of the above, the public loved him regardless – as we do in this country, for a while. Now if you mention his name outside of QL circles, you invariably get a reply along the lines of 'Oh yeah, that stupid electric buggy guy!'.

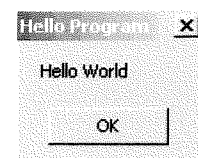
Me, I thought he was well ahead of technology at the time and that's why he had problems. Mind you, he was interested in Wafer Scale Integration at the time when IBM had failed to get it working, and Sinclair employed the guy sacked by IBM. They got it working and were supposed to be producing a computer which was multi processor enabled and could automatically rout out any duff processors in the array (all on one chip) so that only the working ones were used. I remember reading about trials etc, but I never heard anything else – shame really.

The Letter Box

Frank Gutteridge asks 'how many windows users can quickly program to write to screen a little window saying 'Hello World'? Not many I suspect, but for a bit of fun, I did it in raw Windows code, not the C++ Builder or Delphi stuff, so here it is:

```
#include <windows.h>
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    return MessageBox(NULL, "Hello World", "Hello
    Program", MB_OK);
}
```

Not much to it, the output is attached for reference, but it is actually a cheat. A real Windows program to do the same thing is over two pages of code long because a proper Windows program has to define a window, register it and create a message



CAMBRIDGE

Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

MOBILE: 07808 576118

E-MAIL: WNR@COMPUSERVE.COM

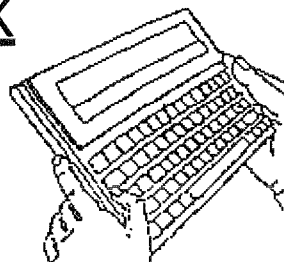
6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL, PC AND MAC.

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.

USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



WE DO REPAIRS, PART EXCHANGES, AND BUY Z88's & PARTS

NEW! CAMBRIDGE Z88 WITH OZ4+512k Internal Ram £150

ALSO AVAILABLE. 512k RAMPACK £40

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	\$180
CAMBRIDGE Z88 COMPUTER	£99	\$130
★ RECONDITIONED Z88	£60	\$90
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac; 6v,500ma)	£10	
TOPPER (PROTECTIVE COVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12

**1/2 PRICE
STARTER PACK
1 - Z88 ★ PLUS
TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!**

FILE TRANSFER TO OTHER COMPUTERS

SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	\$3
PCLINK KIT (For PCs) }	£25	\$35
Z88 TO MAC KIT }	£25	\$35
Z88 TO BBC KIT }	£25	\$35

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

loop. Once all that is done, the window must be physically created and then shown on screen whereupon it will then start to process all the messages that are sent to it, act upon them, and kill itself off when required to. On removal from the system, or when finished, it will not tidy up any resources it has allocated unlike QDOS.

By the way, when I first started Windows programming many years ago, I needed the help of a very large book to get me that far. Oh, if you decide to move the window on screen, or another program overlaps part of it, the bits underneath must be redrawn again when the other program goes away. Fun, not!

And Finally

Linux Format magazine dated November 2001 has a number of pages on the QL written by Simon N Goodwin (where have I heard that name before?) and includes a few emulators, and QDOS stuff on the CD – My old web page is on there as well :o). There are mentions of the Emulators CD from Q-Celt amongst other things as well – so look out for lots of extra business.

Regards, Norman.

PS. I had to write this, I didn't find any bugs in my article this issue !

Simon Goodwin writes:

The latest issue of Linux Format (LXF20, November 2001, out now) contains about three pages of QL and Qdos related material by me including UQLX big-screen and Q60 pictures. It's the latest part of my latest emulation series and I'm glad to have managed to devote as much space to Qdos this time as I did in the Amiga Format one (now online at

<http://www.emuclassics.com/aer/articles/af/>)

This might be of general interest to Quanta/QL Today readers. There's also some information about the ZX80, ZX81 and Ace emulators for Linux, and a plug for Dilwyn's website (among others) and Darren Branagh as source of the emulators CD, which I gather is at last compatible with the Rockridge CD standard so it no longer requires a Microsoft Joliet bodge to get at the long file names. This should make it a lot easier to use on Amigas, Mac 2s, Unix and older Linux distros.

Editor's note: The latest QL Emulators CD should in theory be readable on either Joliet or Rockridge systems, but I am awaiting feedback as I have no Rockridge system on which to test it! The Rockridge version was created using an alpha Windows version of the Linux cdrecord utility, obtained from the internet thanks to

information supplied by Richard Zidlicky, a really good example of the co-operation you get within the QL community.

Al Boehm writes:

Here are some comments on the Sep/Oct issue.

Jochen's article "Devices in QPAC2, QMenu etc. asks for some new names for RomDisq. I don't have any good names but I have some suggestions:

1. Use small letters and Capital letters to differentiate: dev, Dos, ram, Rom.

The Pointer Environment does not distinguish the case for selection keystroke - Jochen.

2. A general way to allot keystroke entries is to use the first letter except when it is already used, then use the second letter. If it is already used, use the third letter. Thus d=dev, o=dos, r=ram, m=rom. With an underline under the keypress letter it works OK.

Yes, but that would mean that the software would have to be re-programmed. This can be taken into consideration for future upgrades, but it could mean that the keystrokes could change depending in which order devices are being loaded - Jochen.

On p51 of "Byfleet 2001, you mention that Bernd Reinhardt author of SERNET had to have a cable made to demo SERNET. Since I am considering using SERNET between Q-emuLator on my laptop and the QXL on my desktop, I am very interested in what are the required pinouts. I hope you can tell me it just needs a null modem (which I already have). I suppose I could query Tony directly but I would guess other people might want to know.

I may be wrong, but hasn't Roy mentioned the Sernet clue in the last issue of QL Today? Maybe Al had not received it by the time he wrote the letter. If not (maybe Roy was planning to do so) then a short summary: the Null Modem cables Roy had bought had no real control lines, they were just bridges. When Tony soldered proper crossed cables, it worked. If we buy a Null Modem cable here in Germany, it is "properly" wired to work with SERNET. Maybe Tony (who seems to have the most knowledge about serial ports and cables thanks to superHermes) could write about it. And, Al, have a look at the blue-covered back-issues, I am pretty sure we had an issue with the pinouts (lots of connectors on the cover) - I am not an expert but crossing TX and RX and the handshake lines should work. - Jochen

Long programs like Herb Shaff's really ought to be on the cover disk. I did note that there was not much room left on this one. I have also encouraged Herb to put all his math/graphic programs on his web site connected via NESQLUG. There is the problem that some people might access them from the site and not buy QL Today. That could be solved by

(1) not putting the accompanying article on the site or

(2) not putting them on the web until some lag period (say 6 months) after publication.

You are right. As we cannot have a cover disk with every issue, the website would be a good place. The article would not need to be on the website, so that would not be a problem, I'd say. Maybe a good place is Dilwyn's QL Today section of his excellent site? - Jochen

This brings up another point, to what degree are programs published in QL Today copyrighted? Or rather who owns the copyright? Since authors are not paid, they could claim they still maintain copyright. A small sentence in the boiler plate could save some trouble later. For example, all published programs are freeware unless specifically noted otherwise. GAD ZOOKS! I am beginning to think like a lawyer!

Doesn't it say so on page 2 of the small print of every issue? We do not want to "take" the copyright, but when we started QL Today, we had to think about potential problems. That's why we put it in. Mainly to avoid to see the whole (or half) contents of the magazine being printed or published elsewhere. It is, of course, no problem, for any author, if he/she wants to publish it anywhere else. A quick note to us is all we want, because it would be pretty pointless to have articles published at the same time in several places. - Jochen

Norman Dunbar writes:

Picture the scene, a very expensive Compaq laptop used by one of our consultants suddenly, and for no apparent reason, loses the up arrow key from the keyboard. The key is still there, just no longer attached. Sometimes they can simply be reattached, but due to the heavy handedness of said consultant, this one has had all its 'bits' broken off and no longer wants to fit back in the slot it came from. Being a nosey sort, when it comes to how things work, I had a look at how the key was attached, and discovered a small rubber 'spring' thing underneath - you've guessed it, Compaq keyboards use a membrane and rubber 'bubble' under the keyboard - just like the QL does. Sir Clive does it again!

Stephen Poole writes:

[..] As far as Linux is concerned, I won't buy commercial magazines to get information, as these are very expensive and contain in excess of 50% advertising. Most written articles in them are just rehashes of publicity anyway, so there is indeed little to be learnt from them. This leaves books on Linux which are even dearer to buy and which seem to get bogged-down in detail. I think it could be of interest for QL Today readers to see how the Linux operating system compares to Qdos, and how Qdos could benefit by adopting Linux-like policies. As far as I am concerned, any technology which gets away from the MS Behemoth-system is worth a brief assessment.

Difficult to judge - let our readers decide! I do not know anything about Linux. We would need to ask somebody who has a good knowledge of both QDOS/SMSQ and Linux. Any volunteers? - Jochen

Many thanks also for Vol 6 Issue 2 of QL Today. On page 24 under QLCF 64, Jerome Grimbert mentions the possibility of assembler-programming in Basic, and Norman Dunbar's recent articles seem to head in this direction too. Is this really conceivable?

Bruno Coativy's article on page 36 seems to cut through a lot of confusion surrounding potential QL systems. When I bought my first JM-QL in 1984 I was told it was a prototype, and I think that this will always be the case for any new QLs. If the Q60 cannot be produced, then we should get QPC, QXL, Q60 and Goldfire designers together, and get them to look into the future for a new collectively planned machine. About 10 years ago and when PCs were running at around 100 Mhz, DEC produced the 500 Mhz alpha chip, which never got into market because no one dared write programs for it! Computer history is full of missed opportunities, so we should grab new technologies as they arise. Admittedly expensive Intel and AMD machines may soon be running at 2 Ghz, but what percentage of the installed PC park can run at such extreme rates and how much software will be optimised to take full advantage of it?

The Q60 already exists in very small quantities, I heard, and I think it is quite unrealistic to get the QXL designer (Miracle Systems) back into the scene. QPC has nothing to do with hardware, and it is improving a lot as you can see from the News. It is mainly GoldFire we're waiting for, isn't it? - Jochen

SMSQ/E and SBasic have been drastically optimised and all QL software can benefit from

this. There is plenty of Free ex-Commercial software which is now available for easy integration into interactives modules. Who knows, if SBasic grows sufficiently from the numerous sources available, it may even become attractive to new computer enthusiasts, allowing commercial houses to develop new innovative programs. If good new software is produced, then the QL system could regenerate itself again. I for one am always looking for new products to spend my money on, which may answer Roy Wood's question as to why do people read QL Today?

There are plenty of products that I could buy, but being on a tight budget I must pick and choose carefully. At the moment I am tempted by CDROM facilities with Qubide, and to a lesser extent by Internet access.

Who said there is no money to be made from the QL? Every time I go to a QL Show I come home with my pockets lightened..... But I am waiting for new QL technological Leaps such as Bruno's Coldfire 333 Mhz QXL, in case the current batch of new QL hardware like the Q60 fails to materialise. At the present moment the QL Scene seems to be a bit dispersed and I only hope that developers can reach some sort of consensus.

Robert Newson writes:

Firstly, I'd like to say that I enjoy QL Today and look forward eagerly to the next edition - it's so good to have a magazine about the QL again.

Also, in a Nov Linux mag (linux Format I think - it's the first time I've bought it) there's an article about emulators (by one Simon N. Goodwin) of the QL, Spectrum, ZX81, ZX80 - can't remember exact details as I've left mag at work.

Here's a few comments about the Sept/Oktober edition of QL Today (Vol 6, Iss 3). Please feel free to edit/ignore any or all of them.

QL Filename Extensions

A few extensions that I know:

_exe as _exec [list had _exc; is this what was intended?]
_lnk file to be Linked (generated by my assembler; formatted to SROFF)
_lst as _lis
_prg Archive program file (ASCII listing)
_pro Archive program file (object [tokenised] format) [can be protected against listing and editing]

We have removed the Windows-related Extension. The letter page is very long this time and we keep it for the future, as this time the list was called "QL Filename Extensions". Thanks for providing them. - Editor

The Ultimate Sin?

I found this an interesting article. I also felt that the QL had design faults - especially noticeable once I had got one and started using it (the native multi-tasking + bundled word processor, Spreadsheet and database were the real attractions to me).

Some of the design I'd love to know who to blame:

1) Lo-res mode (Mode 8) flash - on the ZX Spectrum there was a BRIGHT bit, which effectively gave a choice of 16 colours; the technology to do intensity must have existed at Sinclair, so to have a dim/bright bit would have given the QL effectively 16 colours in "lo-res" mode - it took EGA before the PC had 16 colours - CGA allowed only 4 colours of either Red/Yellow/Green/Black or Magenta/White/Cyan/Blue (if I remember rightly). Who can honestly say they have really used the FLASH bit? (I've sussed that the bits work in pairs: of the flash bits that are set on a line, the odd ones start the flash, the even ones end it. On one half of a timer, all the pixel colours defined between the flash bits are displayed; on the other half, the colour of the first flash bit is held over until the next flash bit. The problem comes if a single flash bit is inserted between two: it now becomes the end of the original flash area, and the end of the area becomes the beginning of the next, etc; similarly changing mode (esp hi- to lo-) causes a great mess.)

2) 32K OS ROM - I believe the original idea was to supply just the basic OS in the ROM with just enough commands (LOAD/RUN/EXEC?) to load the BASIC interpreter or other application; the intention of 32K being to use the short (16bit addresses) mode of the 68K to access the ROM. The BASIC interpreter then got cobbled into the ROM, making it 48K, forcing it to use long (32bit addresses) addressing (with the BASIC actually being "loaded" at boot?). Would it not have been possible to use the -32K short address space (ie the TOP 32k of memory) for the language interpreter (say 16K of it; the other 16K being used by the cartridge slot)?

3) Sound - the manual suggests that the best way to understand the QL's sound is to experiment with the BEEP command. This sounds to me like a cop out in that the [manual] author couldn't give an authoritative explanation of what the parameters actually did. A Maplin kit (for about UKP 10 if I remember rightly) provided a ZX81 with the same sound chip as a BBC! on an expansion board.

4) Serial ports - rather a cludge with the ports

QUANTA

Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our Eighteenth successful year ***

Further details from the Membership Secretary

Bill Newell, 213, Manor Road

Benfleet, Essex, SS7 4JD

Tel. +44(0)1268 754407

or

Visit the Quanta Web Site

<http://www.quanta.uni.cc>

email: quanta_membership@uk2.net

being split into two halves, with one chip (the 8049) handling the input side, and another (the ZX8302) handling the output side. Both the inbound data lines from the serial ports appear to be connected to a single port (and the interrupt line) on the 8049. When the start bit is received it interrupts the 8049 and the data is then clocked in; the port allowed to interrupt the 8049 is indicated by it setting DTR or CTS high – this can be seen if a LED serial port tester is connected: DTR & CTS flash at a rate relative to the baud rate set (very pretty); thus hardware flow control IS essential (esp for received data). Taking Sinclair's delight in using ULAs (makes clones hard) these (latter) moans aren't really surprising.

I found the 128K (77K free after screen, etc taken into account) very usable for my program development: File Editor, Assembler, Machine Code Monitor AND program under development in memory all at the same time with easy switching between them; almost impossible with a 640K PC then.

I have heard it rumoured that the 68008 was chosen as it was in plastic and cheap whereas the 68000 was in ceramic and expensive when the QL was designed. Further, when the QL went into production the 68K was then in plastic and cheaper than the 68008! (Is there any truth to this?)

I believe that the 68008 uses 20 address lines [1M] whereas the 68K uses 24 [16M] – in both cases the full 32 bits are used to specify address in programs, but the extra [high] bits are "ignored" by the processor; thus addresses 0, 1048576, 2097152, 3145728, ... are the same to the 68008, whereas for the 68000 it's 0, 16777216, 33554432, 50331648, ...; the 68020 is [was?] the first of the family to use the full 32 bits to address the "outside world".

Programming in Assembler - Part 9

STR_INSERT isn't quite right. If D0 := length(A1) then it uses STR_APPEND (by jumping directly to the code). This is fine except for one small detail: STR_INSERT returns an error code in D0, whereas STR_APPEND returns D0 as it was. ONLY if both the inserted string (A2) and insertee string (A1) are of zero length will STR_INSERT (with any D0 := 0) return with Z flag set, even though D0 may not be zero. If either string has length, and D0 := length(A2), D0 will return it's original value and the Z flag will be clear, indicating error, even though the insert (append) has succeeded! To correct, I'd suggest:

```
str_insert
    cmp.w  d0,(a1)      ; Are we appending perhaps?
    blt.s  si_doit     ; No, so skip to doing it here
    bsr    str_append  ; Use str_append
    bra.s  si_exit_ok  ; Goto no error return

si_doit
    tst.w  d0          ; Is there anything in D0 ?
    bge.s  si_ok       ; Yes, negatives are bad!
    .
    .
    .
    .
    .
si_next
    dbra   d1,si_imove ; Insert the rest
    movem.l (a7)+,d1/a1-a4 ; Restore those workers
si_exit_ok
    moveq  #0,d0        ; No errors
    rts
```

I prefer to use MOVEQ #0,Dn as opposed to CLR.L Dn for two reasons:

- 1) it's slightly faster (how much my manual didn't say)
- 2) the 0 can be defined in a header file (or at the beginning of the [library] program file) as, say, ERR_OK and so is slightly more obvious as to what is happening (esp if, say, ERR_BP is defined to have the value -15), eg:

```
ERR_OK    equ    0          ; No error
ERR_NC    equ    -1         ; Error Not Complete
...
ERR_BP    equ    -15        ; Error Bad Parameter
...
ERR_BL    equ    -21        ; Error Bad Line
```

(either all of them, or just the ones needed) which then allows:

```
...
si_doit
    tst.w  d0          ; Is there anything in D0 ?
    bge.s  si_ok       ; Yes, negatives are bad !
    moveq  #ERR_BP,d0  ; Bad Parameter
    rts

si_ok
    movem.l d1/a1-a4,-(a7) ; Save those workers
    .
    .
    .
    .
si_next
    dbra   d1,si_imove  ; Insert the rest
    movem.l (a7)+,d1/a1-a4 ; Restore those workers
si_ok_exit
    moveq  #ERR_OK,d0   ; No errors
    rts
```

ERR_OK looks at first glance as an error return, not ok return (it starts with ERR_), so as an alternative you could use:

```
NO_ERR    equ    0          ; No error
```

However, the ERR_OK ensures a standard (ERR_«something») when returning from a function, whether it is an error or OK.

Peter S Tillier writes:

Here are some additions/corrections for the File-name Extensions feature:

- _a C68 Library
- _c C source file
- _h C header file
- _i C68 intermediate file
- _i Intercal source file
- _l Flex input file
- _o C68 compiled object file ready for linking
- _s C68 assembler file
- _y Yacc/Bison source file

- _gz Gzip archive
- _pl Perl source file

- _exe executable file
- _in assembler include file (GST Macro Assembler/QUANTA QMAC)
- _inc ditto
- _mac assembler macro file
- _obj assembled file for linking

- _rex REXX source file
- _rexx REXX source file

- _awk awk source code script
- _sed sed source code script

Hope these are of some use ...

Steve Poole wrote a second letter:

In QL Today, vol 6, Issue 3 of Sept-Oct 2001, Roy Wood relates how people said that they had never got QPAC II working and had given up trying. This is not surprising when you VIEW the various BOOT files on the QPAC II master disk:

BOOT : is ok.

BOOT_qpac2: contains the following lines:

```

180 LRESPR rext$ & 'lng_text_ext'
290 ERT_HOT_RES('t',exec$ & 'qtyp')
300 ERT_HOT_CHP('k',exec$ & 'calendar')
310 ERT_HOT_CHP('c',exec$ & 'calculator')
320 ERT_HOT_CHP('a',exec$ & 'alarm')
330 ERT_HOT_CHP1('w',exec$ & 'typer')
400 ERT_HOT_WAKE('s','sysdef')

```

BOOT_simple : is ok.

BOOT_another: contains the following lines:

```

210 LRESPR 'flp1_qtyp_spell'
270 ERT_HOT_RES ('c','flp1_calculator')
280 ERT_HOT_RES ('k','flp1_calendar')
290 ERT_HOT_RES ('w','flp1_alarm')
310 ERT_HOT_LOAD('d','flp1_QD')
340 EXEC 'flp1_CLOCK'
350 EXEC 'flp1_Sysmon'

```

BOOT_pasion: contains the following line:
160 LRESPR 'flp1_qtyp_spell'

All the above lines suppose that the aforementioned programs are to be found on the disk, but this is not the case. So to run QPAC II, just LOAD each boot program and EDIT them to REMark out the above lines, and SAVE the modified boots onto your backup disk.

QPAC II will then run satisfactorily with one exception: If you select the JOBS menu and then choose SUPERBASIC, the QL first displays a menu, but then freezes and must be RESET.

As I use Romdisk and Superhermes with an extra UK keyboard, the first line of my BOOT program must be:

```
100 ROM_LOAD 'rom1_ipcextuk_bin': TK2_EXT
```

Also, after loading the extended environment, and before loading QPAC II, modify the program line as follows so that the mouse will operate:

```
230 IPCMOUSE: LRESPR rext$ & 'QPAC2'
```

Now QPAC II runs satisfactorily except for one minor quirk. When the button-frame is shown across the top of the screen, double-click on FILES, and the menu of files on flp1_ will be shown. Now quit the menu using ESCape or the Zzz option and you get back to the button-frame. But each time you do this, an extra button called FILES-FLP1_ will be added to the frame. So click on the new button again to get back into the files menu. Then click on the F3-Command button and the Command menu will appear. Clicking on QUIT will get you back to the button-frame and get rid of the extra files button. If you don't click on QUIT to leave the menus, you will rapidly be invaded by numerous files-buttons, and you may get memory fragmentation unless you quit them in strict reverse order to their creation.

So I now have a working QPAC II that I am happy with, but which I only use when I need to run many programs at once.

Bruno Coativity also has a few missing extensions for us:

- _rsp resident procedure (very old days)
- _qwk compressed mail packets

Jens Wildgruber wonders about _lha:

- _lha special LH Arc compressed file

Programming in Assembler - Part 9 / 2

Norman Dunbar

Editor's note: We continue with more useful routines for our library.

Please don't be confused about the "/ 2". Norman numbers the parts on his side without knowing if and how we split them. In order to avoid confusion, we both thought it would be a good idea to use his numbering.

So, if we have to split his numbered parts into two or more sub-parts, we add a sub-part number from now on.

We know these routines are a lot of typing (not only the long BASIC listings commented on the letter pages). However, *reading* them helps a lot to understand them - much more than having them somewhere on a web site. We have seen several times that readers actually follow the listings and comment on bugs, and this is very good - the author will probably like this very much as he sees that his articles are interesting - and we are happy too.

```
*-----*
* NAME          FILE_GET_HEAD
*-----*
* DEPENDENCY    None
* PURPOSE       To read the 64 bytes header for a file. (already open)
* DESCRIPTION    Reads a 64 byte file header for the open file whose ID is passed
*               in A0.L into the buffer (user supplied) whose address is passed
*               in A1.L. This buffer must be at least 64 bytes long !
* INPUTS :
*               A0.L = Channel ID
*               A1.L = Address of 64 byte buffer to put header into
* OUTPUTS :
*               D0 = Error code
*               D1 = Size of header read into buffer
*               A0 = Channel id (preserved)
*               A1 = Address of buffer (preserved)
*               Z flag set if no errors, unset otherwise.
*-----*
file_get_head  movem.l d2-d3/a0-a1,-(a7) ; Save those working registers
               moveq  #FS_HEADR,d0    ; Get trap code
               moveq  #64,d2          ; Buffer size
fgh_rest      moveq  #-1,d3            ; Infinity is a big thing
               trap   #3              ; Do it
               movem.l (a7)+,d2-d3/a0-a1 ; Restore those workers
               tst.l  d0              ; Set flags
               rts                    ; Return to caller
*-----*
```

```
*-----*
* NAME          FILE_SET_HEAD
*-----*
* DEPENDENCY    FILE_GET_HEAD
* PURPOSE       To write a 64 bytes header for a file. (already open)
* DESCRIPTION    Writes a 64 byte file header for the open file whose ID is passed
*               in A0.L from the buffer (user supplied) whose address is passed
*               in A1.L. This buffer must be at least 64 bytes long !
* INPUTS :
*               A0.L = Channel ID
*               A1.L = Address of 64 byte buffer holding the (new) header
* OUTPUTS :
*               D0 = Error code
*               D1 = Size of header written from buffer
*               A0 = Channel id (preserved)
*               A1 = Address of buffer (preserved)
*               Z flag set if no errors, unset otherwise.
*-----*
file_set_head  movem.l d2-d3/a0-a1,-(a7) ; Save those working registers
               moveq  #FS_HEADS,d0    ; Get trap code
               bra   fgh_rest         ; Do rest via FILE_GET_HEAD
*-----*
```

```

*-----
* NAME          PRINT
*-----
* DEPENDENCY    None
* PURPOSE       To send the string at (A1) to the channel in A0.
* DESCRIPTION    This routine prints a QDOS string (word then bytes) to the
*               channel ID passed in A0. The string starts at A1.
* INPUTS :
*               A0.L = Channel ID
*               A1.L = Address of a QDOS format string to be printed.
* OUTPUTS :
*               D0 = Error code
*               A0 = Channel id (preserved)
*               A1 = Address of buffer (preserved)
*               Z flag set if no errors, unset otherwise.
*-----
print           move.l a1, -(a7)          ; Preserve the buffer address
                movea.w ut_mtext,a2      ; Print a string utility
                jsr   (a2)                ; Print it
                move.l (a7)+,a1          ; Restore the buffer address
                tst.l d0                 ; Check for errors
                rts

```

```

*-----
* NAME          LINE_FEED
*-----
* DEPENDENCY    None
* PURPOSE       To send a linefeed character to the channel in A0.
* DESCRIPTION    This routine prints a linefeed character to the channel ID
*               passed in A0.
* INPUTS :
*               A0.L = Channel ID
* OUTPUTS :
*               D0 = Error code
*               A0 = Channel id (preserved)
*               Z flag set if no errors, unset otherwise.
*-----
line_feed      movem.l d1/a1, -(a7)       ; Preserve any registers reqd
                moveq #io_sbyte,d0       ; Send one byte to channel
                moveq #linefeed,d1       ; Byte to send = linefeed
                moveq #infinite,d3       ; Timeout
                trap  #3                 ; Do it
                movem.l (a7)+,d1/a1      ; Restore
                tst.l d0                 ; Set Z if errors
                rts

```

```

*-----
* NAME          INPUT
*-----
* DEPENDENCY    None
* PURPOSE       To obtain some input from the user via the channel ID in A0.
* DESCRIPTION    This routine allows the user to type some input into a buffer
*               (which is part of this subroutine) which allows a maximum of
*               256 bytes to be typed. A channel ID in A0 is used.
* INPUTS :
*               A0.L = Channel ID
* OUTPUTS :
*               D0 = Error code
*               D1.W = Number of characters typed EXCLUDING the ENTER character
*                   if D1.W = 0, user simply pressed ENTER.
*               A0 = Channel id (preserved)
*               A1 = Start of buffer (word count of string user typed)
*               Z flag set if no errors, unset otherwise.
*-----
input          movem.l d2-d3, -(a7)       ; Preserve working registers
                lea   i_buffer+2,a1      ; Our buffer address plus 2
                move.l a1, -(a7)        ; Save it on the stack

```

```

moveq #io_line,d0 ; Input some bytes (inc linefeed)
moveq #256,d2 ; Buffer size maximum
moveq #infinite,d3 ; Inifinite timeout
trap #3

move.l (a7)+,a1 ; Restore buffer pointer
subq.w #1,d1 ; Subtract the linefeed character
move.w d1,-(a1) ; Store length in buffer and set A1
movem.l (a7)+,d2-d3 ; Restore those workers
tst.l d0 ; Did it all work ?
rts

```

```

i_buffer ds.b 256+2 ; 256 chars for input plus 1 word for size.

```

```

*-----
* NAME          JOB_HEADER
*-----
* DEPENDENCY    None
* PURPOSE       Code required to define a QDOSMSQ job header.
* DESCRIPTION    Defines a job header ready to be filled in by the user. All the
*               user has to do is fill in his/her own jobname between the quotes
*               and the assembler will do the rest.
* INPUTS :      None.
* OUTPUTS :     None.
*-----
start          bra.s  prog_start      ; Short jump to program start
               dc.l   0                ; Spare. Use for a serial number ?
               dc.w   $4afb            ; Job identifier must be at location 6
prog_name      dc.w   prog_start-prog_name-2 ; Length of job name
               dc.b   ''              ; YOUR JOBNAME HERE BETWEEN THE QUOTES

prog_start    PUT YOUR CODE HERE

```

```

*-----
* NAME          MEM_ALLOC
*-----
* DEPENDENCY    None
* PURPOSE       Allocate an area of memory on the heap.
* DESCRIPTION    Allocate an area of memory, size as specified in D0.L, and
*               return the address of the allocated area in A0.L. D0 will be
*               set to an error code and the Z flag will be set if no errors
*               occurred, reset otherwise.
* INPUTS :      D0.L = Size, in bytes, of memeoery area to be allocated
* OUTPUTS :     A0.L = Base address of the memory area allocated
*               D0 = Error code
*               Z flag set if no errors, unset otherwise.
*-----
mem_alloc      movem.l d1-d3/a1-a3,-(a7) ; Save working registers
               move.l  d0,d1            ; Space required has to be in D1
               moveq  #MT_ALCHP,d0     ; Set the trap
               moveq  #-1,d2           ; I want it for the current job
               trap   #1               ; Do it
               movem.l (a7)+,d1-d3/a1-a3 ; Restore working registers
               tst.l  d0               ; Set flags
               rts

```

```

*-----
* NAME          MEM_DEALLOC
*-----
* DEPENDENCY    None
* PURPOSE       Deallocate an already allocated area of memory
* DESCRIPTION    Deallocate a previously allocated area of memory, the address of
*               which is passed in A0.L.
* INPUTS :

```

```

*           A0.L = Address of area to deallocate
* OUTPUTS :
*           A0.L = zero to avoid using the memory again !
*           D0 = Error code
*           Z flag set if no errors, unset otherwise.
*
-----
mem_dealloc  movem.l  d1-d3/a1-a3,-(a7) ; Save working registers
             moveq   #MT_RECHP,d0    ; Set the trap
             trap    #1              ; Do it
             movem.l (a7)+,d1-d3/a1-a3 ; Restore working registers
             suba.l  a0,a0           ; Blank the memory address
             tst.l   d0              ; Set flags
             rts

```

```

*
-----
* NAME      SCR_MODE
*
* DEPENDENCY  None
* PURPOSE     Check the mode & set if required
* DESCRIPTION  Checks for the mode passed in D0 and if not correct, change to
*              that mode.
* INPUTS :
*           D0.B = 4 or 8 for required mode
* OUTPUTS :
*           D0 = Error code
*           Z flag set if no errors, unset otherwise.
*
-----
scr_mode     move.l   d1-d2/d7/a3,-(a7) ; Save working registers
             move.b   d0,d7            ; Save required mode
             cmpi.w   #4,d7           ; Is mode 4 required ?
             bne.s    scrm_8          ; Nope.
             clr.b    d7              ; Mode 4 requires 0
scrm_8       moveq    #mt_dmode,d0
             moveq    #-1,d1          ; Read current mode
             moveq    #-1,d2          ; Read current display type
             trap     #1              ; Do it
             tst.l    d0              ; Did it work ?
             bne.s    scrm_exit       ; No, bale out
             cmp.b    d1,d7           ; Are we in the correct mode ?
             beq.s    scrm_exit       ; Don't set mode if already correct (Z
*           ; set = no errors & D0 = zero)
             moveq    #mt_dmode,d0
             move.b    d7,d1          ; Get the required mode from D7
             trap     #1              ; Set mode
             move.l   (a7)+,d1-d2/d7/a3 ; Restore working registers
             tst.l    d0              ; Set Z flag if no errors
scrm_exit    rts

```

```

*
-----
* NAME      CLS
*
* DEPENDENCY  None
* PURPOSE     To clear a screen/console channel.
* DESCRIPTION  Clears the screen channel whose ID is supplied in A0.
* INPUTS :
*           A0.L = channel ID
* OUTPUTS :
*           D0 = Error code
*           A0.L = channel ID (preserved)
*           Z flag set if no errors, unset otherwise.
*
-----
cls          move.l   d1/d3/a1,-(a7) ; These will be corrupted by SD_CLEAR
             moveq    #sd_clear,d0  ; CLS (SD_CLEAR defined in GWASL)
             moveq    #-1,d3        ; Infinite timeout
             trap     #3            ; CLS the window
             move.l   (a7)+,d1/d3/a1 ; Restore corrupted registers
             tst.l    d0            ; Set Z flag if all ok, else not set
             rts

```

```

*-----
* NAME          SCR_PAPER
*-----
* DEPENDENCY    None
* PURPOSE       Set the PAPER colour for the given channel ID.
* DESCRIPTION    Sets the paper colour for the screen channel whose ID is passed
*               in A0, to the colour code supplied in D0.W.
* INPUTS :
*               D0.W = colour code for paper colour
*               A0.L = Channel ID.
* OUTPUTS :
*               D0 = Error code
*               A0.L = channel ID (preserved)
*               Z flag set if no errors, unset otherwise.
*-----
scr_paper      move.l  d1/d3/a1,-(a7) ; These will be corrupted by SD_SETPA
               move.w  d0,d1      ; Get the paper colour
               moveq   #sd_clear,d0 ; CLS (SD_SETPA defined in GWASL)
               moveq   #-1,d3     ; Infinite timeout
               trap    #3         ; Set PAPER colour (but not STRIP)
               move.l  (a7)+,d1/d3/a1 ; Restore corrupted registers
               tst.l   d0         ; Set Z flag if all ok, else not set
               rts

```

```

*-----
* NAME          SCR_PAPER_SB
*-----
* DEPENDENCY    SCR_PAPER
* DEPENDENCY    SCR_STRIP
* PURPOSE       Set the PAPER & STRIP colour for the given channel ID.
* DESCRIPTION    Sets the paper & strip colour for the screen channel whose ID is passed
*               in A0, to the colour code supplied in D0.W. Works like
*               SuperBasic's PAPER command.
* INPUTS :
*               D0.W = colour code for paper & strip colour
*               A0.L = Channel ID.
* OUTPUTS :
*               D0 = Error code
*               A0.L = channel ID (preserved)
*               Z flag set if no errors, unset otherwise.
*-----
scr_paper_sb   move.w  d0,d1      ; Save the colour between calls
               bsr    scr_paper   ; Set the paper colour
               bne.s  spsb_exit   ; Tets for errors
               move.w  d1,d0      ; Get the colour code again
               bsr    scr_strip   ; Set the strip colour
scsb_exit      rts

```

```

*-----
* NAME          SCR_INK
*-----
* DEPENDENCY    None
* PURPOSE       Set the INK colour for the given channel ID.
* DESCRIPTION    Sets the ink colour for the screen channel whose ID is passed
*               in A0, to the colour code supplied in D0.W.
* INPUTS :
*               D0.W = colour code for ink colour
*               A0.L = Channel ID.
* OUTPUTS :
*               D0 = Error code
*               A0.L = channel ID (preserved)
*               Z flag set if no errors, unset otherwise.
*-----
scr_ink        move.l  d1/d3/a1,-(a7) ; These will be corrupted by SD_SETIN
               move.w  d0,d1      ; Get the ink colour
               moveq   #sd_clear,d0 ; CLS (SD_SETIN defined in GWASL)
               moveq   #-1,d3     ; Infinite timeout
               trap    #3         ; Set INK colour

```



```

move.l (a7)+,d1/d3/a1 ; Restore corrupted registers
tst.l  d0              ; Set Z flag if all ok, else not set
rts

```

```

*-----*
* NAME          SCR_STRIP
*-----*
* DEPENDENCY    None
* PURPOSE       Set the STRIP colour for the given channel ID.
* DESCRIPTION    Sets the strip colour for the screen channel whose ID is passed
*               in A0, to the colour code supplied in D0.W.
* INPUTS :
*               D0.W = colour code for strip colour
*               A0.L = Channel ID.
* OUTPUTS :
*               D0 = Error code
*               A0.L = channel ID (preserved)
*               Z flag set if no errors, unset otherwise.
*-----*
scr_strip      move.l  d1/d3/a1,-(a7) ; These will be corrupted by SD_SETST
               move.w  d0,d1         ; Get the paper colour
               moveq   #sd_clear,d0  ; CLS (SD_SETST defined in GWASL)
               moveq   #-1,d3        ; Infinite timeout
               trap    #3            ; Set STRIP colour (but not PAPER)
               move.l  (a7)+,d1/d3/a1 ; Restore corrupted registers
               tst.l   d0            ; Set Z flag if all ok, else not set
               rts

```

```

*-----*
* NAME          COLOURS
*-----*
* DEPENDENCY    None
* PURPOSE       Define names for the various QDOSMSQ colours
* DESCRIPTION    Not really a subroutine, but a set of equates which define
*               meaningful names for the 8 colours available on a 'standard'
*               QDOSMSQ machine.
* INPUTS :      None
* OUTPUTS :     None
*-----*
black          equ    0
blue           equ    1
red            equ    2
magenta        equ    3
green          equ    4
cyan           equ    5
yellow         equ    6
white          equ    7

```

Finally

Ok, so there you have a few of my favourite routines, all you need now is a librarian to sort them out for you. Ok, I give up, here is one for you - this is very basic and not super at all (sorry for that pun) it is up to you to expand on this if you want.

Some suggestions would be :

Make PE aware?

Add better/some error trapping.

Save the dependencies so that the user need

not enter them manually.

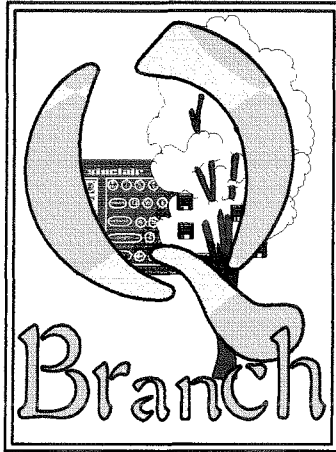
Check what has just been entered with what has already been entered to avoid duplications.

Reduce the number of file open/closes etc (On the Library file)

Convert to assembler - Ha, now you're quaking in your boots!

(I have omitted line numbers from the following listing. This makes it easier for Jochen to get the code all in one line!)

Well, thanks a lot - this should really help (and my reply to this helps getting a neat break to the next page - Jochen :-)



Text 87

£ 79.00

Typset 94 £ 29.00
 Fountext 94 £ 39.00
 2488 drivers £ 29.00
 Epson ESC/P2 drivers
 £ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handling the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Q's Christmas Message

Looking for that elusive Christmas present for the QL obsessed? Q Branch have a few, sought after, hardware items. We have a couple of Auroras, Qubides and even Gold and Super Gold Cards. Call us now while stocks last. Marcel has been working away at QPC2 and a new version may be out by the time you read this - call or email for details. We still have a QXL II in stock and we are negotiating to buy a few membranes from a user who has had them in store for a while. Call us if you need one.

There are also two complete Q 40 Systems available

Most of all we wish all of our customers a Merry Christmas and a Happy New Year.

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
 (Includes Pointer and non-pointer driven versions)
 (P.E. versions need Hot_text, WMAN and PTR_GEN or SMSQ/E to run)
 Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

Convert text files from QL to PC formats and much more!

Only £ 10.00 Now with HTML support!

QL Rhymes £ 10.00
 Rhyming Dictionary

Spelling Crib : PD program £ 1.50 + SAE
 or Free if you buy all three programs

New Program !

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v3.00	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

We are currently out of stock of the SuperBasic Reference Manual
 Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
 compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages!
£ 40.00
 + postage

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577

Mobile +44 (0) 7836-745501

email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

Feeling out on a limb?

Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

Demo version £ 1.50 + postage
Full Registered version £ 18.00



Version 2.03 available now !

SMSQ/E

Gold Card / Atari / QXL Version

£ 69.00

Various Atari versions : call for details

QPC 2 full colour version!

£ 82.00 (£ 67.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 35.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Special offer !

Get Cueshell for only £15.00 with any copy of SMSQ/E
or QPC 2 upgrade !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II £ 90.00

Recycled superHermes £ 60.00 *

Recycled Gold Card £ 55.00 *

Recycled Aurora £ 75.00 *

Qubide £ 55.00 *

Qplane £ 5.00

Aurora cables £ 3.00

Aurora rom adaptor £ 3.00

'Arfa Braquet' £ 8.00

'Son of Braquet' £ 18.00

The 'Braquet' £ 16.00

MC plate £ 6.50

2 x Q40s available !

Call for details !

* when available.

14" and 15" monitors for the Aurora - Call.

Q Branch Programs

The Knight Safe 3 £ 35.00

upgrades from previous versions £ 5.00

Q - Route v2.00 £ 25.00

Route finding programme

Q - Count £ 25.00

Pointer driven home accounting

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

QPC 2 v2 FULL VERSION AVAILABLE NOW

New Version Out Soon ! Call for details !

Qubide upgrades to version 2.01 £ 8.00



We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.



Q Branch

```

CLS
Output = 3
INPUT 'Library name : ' LibraryName$
INPUT 'Output file name : '; Output$
OPEN_NEW #Output, Output$
:
REPEAT main_loop
  INPUT 'Routine name (ENTER to quit) : '; Name$
  IF (Name$ = '')
    EXIT MainLoop
  END IF
  ExtractName Name$
END REPEAT MainLoop
:
CLOSE #Output
PRINT "Done."
STOP
:
:
DEF PROCEDURE ExtractName(ReqdName$)
  LOCAL A$, Library, FoundName$
  Library = Output + 1
  OPEN_IN #Library, LibraryName$
  REPEAT LibLoop
    IF (EOF #Library)
      EXIT LibLoop
    END IF
    INPUT #Library, A$
    IF (A$(1 TO 6) == "* NAME")
      FoundName$ = A$(17 TO)
      IF (FoundName$ == ReqdName$)
        PRINT "Found subroutine : " & ReqdName$
        GetDependencies(Library)
        ExtractCode(Library)
        CLOSE #Library
        RETURN
      ENDIF
    END IF
  END REPEAT LibLoop
  PRINT "Cannot find : " & ReqdName$
END DEFINE ExtractName
:
:
DEF PROCEDURE GetDependencies(Channel)
  LOCAL A$
  REPEAT DependLoop
    IF (EOF #Channel)
      RETURN
    END IF
    INPUT #Channel, A$
    IF (A$(1 TO 12) == "* DEPENDENCY")
      IF (A$(17 TO 20) == "None")
        PRINT "No dependencies"
        RETURN
      END IF
      PRINT "Dependency required : " & A$(17 TO)
    END IF
    IF (A$(1 TO 9) == "* PURPOSE")
      RETURN
    END IF
  END REPEAT DependLoop
END DEFINE GetDependencies
:
:
DEF PROC ExtractCode(Channel)
  LOCAL A$
  REPEAT FindCodeLoop
    IF (EOF #Channel)
      RETURN
    END IF

```

```

INPUT #Channel, A$
IF (A$(1 TO 5) == "*-----")
  EXIT FindCodeLoop
END IF
END REPEAT FindCodeLoop
REPEAT WriteCodeLoop
  IF (EOF #Channel)
    RETURN
  END IF
  INPUT #Channel, A$
  IF (A$(1 TO 5) == "*-----")
    EXIT WriteCodeLoop
  END IF
  PRINT #Output, A$
END REPEAT WriteCodeLoop
PRINT "Extracted."\\
END DEFINE ExtractCode

```

So how does this lot work?

After asking for your details etc, it simply enters a loop asking you for the next routine to be extracted. This name is passed to ExtractName which opens the library file and scans it looking for all those lines which start '* NAME'. Once it finds one, it tests to see if this line includes the name you are looking for.

Note that this version of the program assumes you are using EXACTLY the same format in your comments as I am above and as per the following description:

Column 1 = An asterisk, the comment marker for most assemblers I have used.

Column 2 = A space.

Columns 3 to 16 = Parameter name, eg NAME, DEPENDENCY etc

Columns 17 onwards = Parameter details etc.

If the name found is the same as the one you requested, the dependencies are extracted and listed. You are advised to note these dependencies and enter them as the next routine to extract. Try not to duplicate names etc as the program doesn't test for duplicates.

Once all dependencies have been listed, The code is extracted and written to the output file.

A sample session follows:

```

Library name : Win1_GWASL_Library_lib
Output file name : Win1_source_MyNextProject_asm
Routine name (ENTER to quit) : Colours
Found subroutine : COLOURS
No dependencies
Extracted.

```

```

Routine name (ENTER to quit) : Scr_paper_sb

```

Found subroutine : SCR_PAPER_SB
Dependency required : SCR_PAPER
Dependency required : SCR_STRIP
Extracted.

Routine name (ENTER to quit) : Scr_paper
Found subroutine : SCR_PAPER
No dependencies
Extracted.

Routine name (ENTER to quit) : SCR_STRIP
Found subroutine : SCR_STRIP
No dependencies
Extracted.

Routine name (ENTER to quit) :
Done.

So there you have it and now you can enhance it as required to suit your own purposes. Remember, my version expects the comments to be in the format given above. Additionally, no comments are written to the output file but you can easily amend the code in ExtractCode to do the needful. Enjoy.

Notes on Norman Dunbar's Assembler Parts 8 & 9

George Gwilt

PART 8

Stung by the suggestion in Part 9 of Norman Dunbar's articles on Programming in Assembler that there might be "more bugs" in Part 8 of the series, I took time to read it thoroughly. I must admit that years ago I copied and used the routines for plotting points as described by Pennel in his book, Assembly Language Programming. These routines are of course similar to, and doing the same thing as, Norman's so I'm afraid my glances at Part 8 were initially rather cursory. Another reason for my not delving too deeply at first is that the screen content for the extended colours on my Q40 is entirely different from that discussed by Norman. Indeed plotting pixels when the colour driver GD2 is in operation is a good deal simpler than for Mode 4 or 8. I found this when extending some of the routines in the Turbo_TK_DEMOS which deal with the larger and more colourful screens. However, I must now comment on my findings re Part 8.

1. At the bottom of Page 30 the address of the word containing the pixel x, y is given as
 $\text{screen_base} + (\text{y} * \text{screen_width}) + \text{INT}(x / 4)$.
If x is even this gives the correct answer. However if x is odd (assuming Mode 4) the answer is wrong. For Mode 4 strictly you should replace
 $\text{INT}(x / 4)$
by
 $2 * \text{INT}(x / 8)$
to get the correct answer.

To suit both Mode 4 and Mode 8 the expression could be

$\text{INT}(x / 4)$ rounded down to the nearest even.

A glance at the code for "calc" on Page 35 shows that the rounding down is indeed done, by the instruction

```
bclr #0,d4
```

so all is well in the end.

[ND] - I shall bow to George's knowledge on the arithmetic involved, and say that I'm glad it all worked out in the end. Hooray for 'bclr' instructions, where would we be without them?

2. In explaining "extop" on Page 30 Norman says that it "gets presented with the channel definition block's address in A0". It might be important in some cases to know that this is not true if the Pointer Environment is installed. When the Pointer Environment is installed it changes all CON/SCR channels, setting an extra \$30 bytes to each. The effect is that if you look at the contents of true block address + \$32 you do not find the screen base because that is recorded \$30 bytes further on! This is where SD_EXTOP is very useful, for what it sets in A0 is the channel address plus \$30 if the Pointer Environment is there and the true address otherwise.

[ND] In my QPAC2 docs, I seem to remember seeing details of the extension to the normal SCR/CON channel definition blocks. The bit I liked most is the fact that offsets to the new information would always be negative to A0 while the normal offsets would still apply. I think Tony Tebby made a wise decision, because the new way didn't break any existing programs which used the EXTOP routines. Still, it is worth remembering about the extra bytes.

3. In talking about Mode 8 usage on Page 31, Norman says that the red bits are even and the blue bits odd. In fact it is the other way round. (This is the sort of thing I'm always getting wrong

myself so perhaps it's a failing of Assembly programmers.)

[ND] George is, as ever, correct. My diagram shows the red bits in positions 1,3,5 ... and the blue bits in 0,2,3 ... - I can't tell odd from even, left from right, up from down etc.

4. At the bottom of Page 31 the diagram shows a bit pattern with two flash bits and no green bits. Over the page the last two lines in the table should be:

32	011	magenta
10	010	red

[ND] Once again, George is correct. I have 'stripped out' the green bit rather than the flash bit for the bottom two entries in the table. This binary stuff is not easy you know!

5. Before I come back to "plot_init" I want to point out that on Page 36 the comments on the first two lines of "extop" are transposed. The screen base is at \$32(a0) and the scan_line length is at \$64(a0) and not vice versa.

[ND] George is correct again. The comments are indeed swapped over.

6. I regret to say that "plot_init" does not work on my Q40, or indeed under SMSQ/E. I have had this trouble before. The problem seems to be with the assumption that you can use channel #0 to find information via SD_EXTOP in programs other than the master basic. If you try that in an assembly program the program stops at the EXTOP trap #3. The solution is to supply to SD_EXTOP the ID of a channel used in the program or of one opened specially.

I will end with an example of a section of code (which did work on my Q40) to replace Norman's (which of course will certainly work on non SMSQ/E systems).

This subroutine uses "file_open" and "file_close" from Norman's suggested library of subroutines described in Part 9. Note also that "plot_init" returns with D0 negative for an error and positive, not zero, otherwise.

[ND] This is strange as I tested on my QPC setup and all was well. However, hoping that

Corrected plot_init

```

plot_init lea      con,a0
          bsr.s   file_open      Open a channel in place of #0
          bne.s   done           If an error - exit
          lea     scr_base,a1
          lea     extop,a2
          moveq   #sd_extop,d0
          moveq   #-1,d3
          trap    #3
          tst.l   d0
          bne.s   done
          move.l  a1,-(a7)       Save A1 (see PS below)
          bsr.s   file_close     Close the channel
          movea.l (a7)+,a1       Restore A1
          move.w  d1,-(a7)
          moveq   #mt_inf,d0
          trap    #1
          move.w  (a7)+,d1
          andi.l  #$ff0ffff,d2
          cmpi.l  #$31003034,d2
save      move.w  d1,(a1)

done      rts

too_old   move.w  #128,d1
          bra.s   save

extop     move.l  $32(a0),(a1)+ Set the screen base
          move.w  $64(a0),d1   Set the scan_line length
          moveq   #0,d0
          rts

con       dc.w    3
          dc.b    "CON"

in        flp1_files_sym_lst   See IN/LIB
lib       flp1_files_bin      . . below

```

channel #0 has an id of zero is asking a bit much these day. It has to be that way on a normal QL, but with all the new stuff around, things are obviously different.

IN/LIB

Included in the GWASL pack is the program SYM_BIN. This program turns a _SYM file from its internal format into a readable _SYM.LST file which contains a list of EQUates giving the value of labels used in the _ASM program which assembled to produce the _BIN and _SYM files in the first place. Thus if all the FILE_OPEN and FILE_CLOSE subroutines are put together in one file called "files_asm" and assembled, and SYM_BIN is used on the resulting _SYM file, we get the two files:

files_sym_lst	Output of SYM_BIN
files_bin	Output of GWASL

By including the first file, by "in", just before the second is inserted by "lib", we ensure that the

RWAP SOFTWARE

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Q-Help v1.05 £10

Q-Index v1.04 £5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.

The PD toolkits referred to are available for £2.

Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and find out the commands which relate. Launch Q-Help for further info on the chosen command.

Sidewriter v1.03 £10

Produce landscape printouts of Easel/Qspread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix to laser printers. A most useful utility by Dilwyn Jones - you know it must be easy to use.

ProForma ESC P2 Drivers v1.03 £8

New improved colour and monochrome printer drivers, providing up to 720dpi for all programs written for use with ProWesS, such as LineDesign and Paragraph. Works on all Epson inkjet printers which support binary mode compression (740, 850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26 £20

Genealogy For Windows £50

Store your family tree for posterity. Add individuals with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family. QL version now supports FileInfo II and QMenu as well as allowing you to link both male and female trees. Sample tree of the Royal family since 1066 included. PC version is event driven - enter the details as they appear in documents and it generates the tree from these. QL data and GEDCOM can be transferred to the PC version. Upgrade to latest PC version (v5.21) for £8 Both programs easy to use and complete with a step by step tutorial.

** QL USERS upgrade to PC version for £25 ONLY **

D-Day MKII v3.04 £10

Grey Wolf v1.8 £8

War In The East MKII v1.24

(Upgrade Only) £5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while. Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

Image D v1.03 £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

SBASIC SuperBASIC Reference Manual £40

Updates £6 each. £10 for 2 (Current Version - Rel 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date with regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note: Price for the book does not include post & packing).

QL Cosmos v2.04 £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars and planets for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

Q-Route v2.00 £25

Upgrade from v1.xx £5

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

Flashback SE v2.03 (Upgrade only) £5

The ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately only available as an upgrade from the original version (original still available from Sector Software).

Return To Eden v3.08 £10

Nemesis MKII v2.03 £8

The Prawn v2.01 £8

Horrorday v3.1 £8

West v2.00 £5

The Lost Kingdom of Zkul v2.01 £5

A wealth of QL adventures - mainly text only. Save the Galaxy from the ambitions of the evil dictator Nemesis.

Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday.

Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.

Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.

Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.

Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.

All six adventures are available together for only £25.

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

Open Golf v5.20 £8

QuizMaster II v2.07 £5

Stone Raider II v2.00 £5

Hoverzone v1.2 £5

Deathstrike v1.5 £5

Flightdeck v1.05 £10

The latest maps for Q-Route. Maps of various areas of Britain have been created by cutting them out of Big Britain Map - they will use less memory and can contain more detail. Areas covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England. Latest version of Q-Route is recommended.

Britain.map v1.11 £2

Big Britain Map (needs 2MB) v2.03 £5

Various Britain Area Maps (ask for details) £2 ea.

Ireland Map v1.00 £5

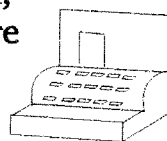
Belgium Map v1.01 £2

Catalonia Map v1.03 £2



RWAP Software, 7 Common Road,
Kinsley, Pontefract, West Yorkshire
WF9 5JR

TEL: 01977 614299



Cheques in £sterling
payable to 'R.Mellor'

* Also known as Trading Accounts

labels "file_open" and "file_close" will have the correct values. Thus "bs.rs file_open" will branch to the correct place in the binary file "files_bin". It is of course assumed here that the library files are kept on FLP1.

[ND] I like this! I never knew that GWASL was able to include binary files, and I always did wonder what SYM_BIN was used for. Now I know. This will make 'modular' programming a bit easier.

PART 9

As a sort of PS I should perhaps mention that I did read Part 9 and found only one possible snag. This occurs on Page 26 where Norman introduces FILE_CLOSE. I think it would be wise to preserve register A1 to be on the safe side. I seem to remember having to do this myself on one occasion when a program refused to work.

In Jochen Merz's QDOS manual it has (about IO_CLOSE) the comment concerning registers on exit from the routine:

A1 ???

[ND] Having looked at my docs I can see that preserving and restoring A1 would most likely be a good idea. I have a feeling I fell over this problem years ago when I had a Trump Card and a Gold Card. Some routine or other in DJToolkit worked on the Trump Card, but not on the Gold Card and it was all down to A1 being trashed. I checked the source for DJToolkit and found that I fixed this on the 19th of July 1993, over 8 years ago! (The bug was in FS_POSRE and not in IO_CLOSE after all.)

[ND] I'm glad the George takes the time to study what I write and point out the obvious errors, and some not so obviouse. Thanks.

The DEMO VERSIONS Cover Disk 2

With Volume 6 Issue 4 of QL Today, we bring you a second disk with demonstration versions of seven commercially available QL software packages. Most of these are fully working versions of the programs, but with some restrictions on their use (e.g. saving or printing may be disabled, or the programs may only work with restricted file sizes or numbers of files). If you like these demo versions, please consider buying the full versions of the programs from the QL software traders. The more they sell, the more they are likely to produce!

Many of the programs require pointer environment (built into SMSQ/E of course, and the files ptr_gen, wman and hot_rext for QDOS), Menu Extension (MENU_REXT supplied with many commercial QL programs) or Toolkit 2 which is built into most modern QL systems.

The seven demo programs are all supplied in zipped file format. This means that they have been packed and compressed using a program called ZIP to allow more programs to be supplied on this disk. You need a copy of the Unzip program to decode these files, so (thoughtfully!) I have provided a copy of that program along with a short SuperBASIC program called BOOT which you should LRUN

to start decoding the seven program zip files. You will need a separate blank, formatted disk for all 7 programs (or as many of them as you plan to try out). Put the cover disk (or preferably a backup copy of it) in FLP1_ and the blank, formatted disk in FLP2_. If you have only a single floppy drive system as I do, you will need to unzip the files to a ramdisk or hard disk first, then transfer them back to a floppy disk.

The program asks you to select one of the seven programs by entering the number shown on screen (type in a number from 1 to 7 and press ENTER). Then it asks you to enter the name of the drive holding the zipped file (e.g. FLP1_), then the name of the drive to unzip it to (e.g. RAM1_ or FLP2_) and finally the name of the drive holding the UNZIP program itself (e.g. FLP1_). Unzip will load and attempt to decompress the files. This may take some time, depending on the speed of your QL system.

QL Unzip is maintained by Jonathan Hudson and full documentation, up to date file sets etc can be obtained from his website on www.daria.co.uk

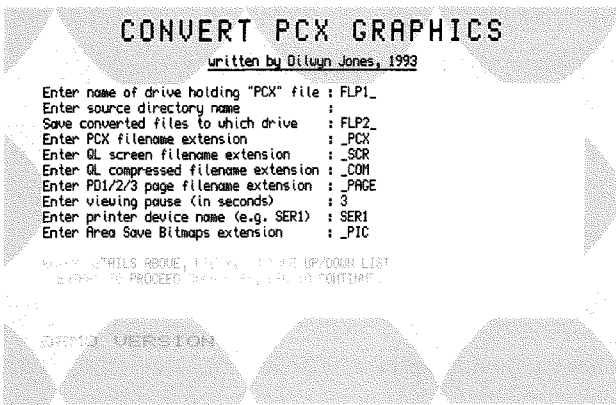
CONVERT-PCX demo version

Author: Dilwyn Jones

Price: £10 from Q-Celt Computing (or part of bundles)

Documentation: PCX_DOC, UPDATES_DOC (Quill files)

Requirements: Expanded memory



Please read the DOC files on the disk for fuller details of how the program operates.

Convert-PCX was written to allow the transfer of bitmapped clipart files in PCX (PC Paintbrush) format to the QL. Convert-PCX should be used in conjunction with a PC to QL file transfer utility such as Discover.

This version of the program handles monochrome (black and white) pictures and 16 colour pictures. It is not yet capable of handling 256 colour files or multi-bit layouts.

Convert-PCX can offer various QL end formats. Pictures which will fit onto the QL screens are saved as normal QL screen pictures for maximum acceptance by other QL software. Screens can be saved uncompressed (should suit every QL program capable of loading screens) or compressed in Image Processor format, which uses far less storage space on a disk. If you have DJC's Screen Compression program, this format can be converted into most other formats used on the QL.

There are several options for pictures which are larger than QL screens. Such pictures can be shrunk until they fit the screen. They can also be saved as a series of consecutive screens or a single section of the large picture can be cut out and saved, or a large picture can be saved as a page suitable for loading into the Page Designer programs (version 1, 2, or 3!) either as compressed pages or uncompressed pages. Pictures can also be saved in the pointer environment PIC file format.

The program can also be used as a PCX file viewer if you wish.

The program operates pretty quickly, especially on fast hardware such as the Gold Card. Speed is reasonable on older QL setups thanks to the use of machine code routines linked to the QLiberated BASIC program.

Convert-PCX requires a memory expansion of at least 256k to operate successfully. While the program itself is fairly short, it requires a great deal of memory to load the original file, decom-

press it and turn it into a QL file.

This version is a demo version which is almost fully working. It allows you to view PCX files, but not to save them in QL converted form, so it acts as a good example of how the program works, and also forms a nifty PCX file viewer. This version is called (surprise surprise!) PCX_DEMO_obj, and it can be started with the EXEC or EXEC_W command.

D-Day Mk II

Author: Games Workshop company, updated by Rich Mellor

Price: £15.00 from RWAP Software

Documentation: MANUAL_DOC (Quill file) and README_TXT (plain text)

Requirements: 256K Min. memory, 512x256 QL screen resolution



This is the public domain demo version of D-Day Mk II, a truly excellent wargame for the Sinclair QL.

D-Day is a simulation of tactical combat in the Second World War. The PD version is a one player game - you take command of the Allies (British and American forces), and the Computer takes command of the Axis (German forces). Each side is allocated a number of units which they have to manipulate in order to achieve their aim. These units are capable of moving around the map, firing at, or going into close combat with other units. Each unit has five attributes, which represent attack, defence, movement, range and damage. These limit the amount that a unit can do in a round and will be further defined below. The game should run on all implementations of the QL with a minimum of 256K memory (including the various emulators which are available). However, if your QL (or emulator) supports a higher screen resolution than 512x256 pixels then you will need to ensure that the system is set up to run in 512x256 pixel mode. It should however run wherever the screen address starts (normally \$20000 (131072)).

If you use this program on a fairly quick QL, (eg. Super Gold Card), then you may like to use the command SLUG 2 to slow down the system so that you can watch the computer make its moves.

Obtaining the Full Version

The full version of D-DAY MKII has several interesting features over and above the Public Domain version:

- The computer can play either the Allies or the Germans
- You can also play against another human player.
- You can play with defined start-up positions of 15 or 50 unit armies
- You can select the size of the armies (between 10 and 50 units) and place them on the map as a starting position
- You can load saved games (this allows play by mail).

The full version is only available commercially at a cost of £15 from Rich Mellor at RWA Software.

DISA disassembler demo version 2e03

- Author:** Jochen Hassler
Price: £31 from QBranch (note: version 3 now available)
Documentation: DISADemo2e03.txt (plain text file)
Requirements: Pointer environment, Expanded memory, Menu Extension

```

F1 STATUS F2 DISA 2e03 0Jo Zc ESC
dc.b dc.u dc.l prel grel defp text code ascii F3 FILES F4 SEARCH F5 EDITOR
0000 43FA0034 lea L0036(pc),a1 C..4
0004 6000009A bra L0000 £...
0008 613C bsr.s L0046 a<
000A 4A00 tst.l d0 Ji
000C 6622 bne.s L0030 f*
000E 2F08 move.l a0,-(a7) /
0010 34780112 moveq.u #0000112,a2 4x..
0014 4E92 jsr (a2) N.
0016 205F moveq.l (a7)+,a0 -
0018 7200 moveq #000,d1 r.
001A 7400 moveq #000,d2 t.
001C 5943 subq.u #004,d3 YC
001E 6708 beq.s L0028 g.
0020 22369008 move.l #00(a6,a1.l),d1 6..
0024 5543 subq.u #002,d3 UC
  
```

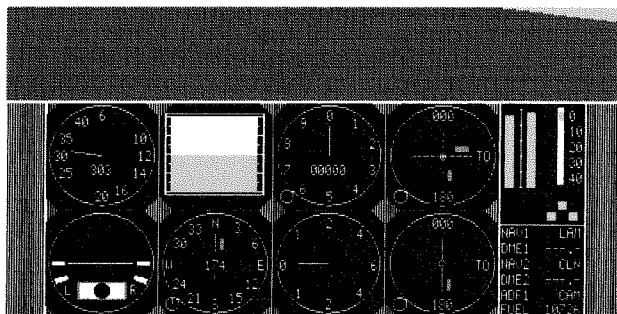
The DISA Demo version 2e03 (English version) is the same as the full version. It is NOT possible however to save your results. This is true both for a DISA listing and an Index file. With the full version, you get:

1. The manual (english), 46 pages DIN A5, describing getting started, the DISA idea, The menu items, tips and tricks, the pointer environment and so on.
2. The diskette containing the program and examples.

To start the demo version, ensure pointer environment is installed and execute the main program DISA2e03. When the main display appears, ensure that the CODE item is highlighted. Then click on the FILES item (or press F3) and when the LOAD/SAVE dialogue box appears, HIT (left mouse click) on the Files item and type in the name of a machine code file, or in the Memory item, type in an address in memory to disassemble. If you have QMenu installed and prefer to use the File_Select menu for filenames, DO (right mouse click) on the Files item. Once it has loaded the file, press ESC and the program will show the disassembled code. Scroll up and down through the listing and try out the available options via the commands at the top of the screen.

FLIGHTDECK

- Author:** Bernard H. Denchfield, Deltasoft
Price: £10 from RWA Software
Documentation: MANUAL_DOC (Quill file)
Requirements: None



Flightdeck is a flight simulator program for the Sinclair QL. Written mainly in machine code, it allows you to "fly" a twin-engined passenger jet airliner, and provides high resolution shaded 3D views of the "world" outside. A database of 25 major UK airports and over 200 navigation beacons is supplied. The "aircraft" is fully equipped with twin VHF omnidirectional range and distance measuring equipment (VOR/DME), automatic direction finding (ADF), and an instrument landing system (ILS).

The demo version is time limited - you are limited to just a few minutes-worth of flying time. The full version of FlightDeck allows you to change the world around which you are flying, including the ability to add additional 3D figures.

You may like to configure the program, using the CONFIG_EXE program supplied. Simply EXEC_W flp1_CONFIG_exe and then follow the prompts.

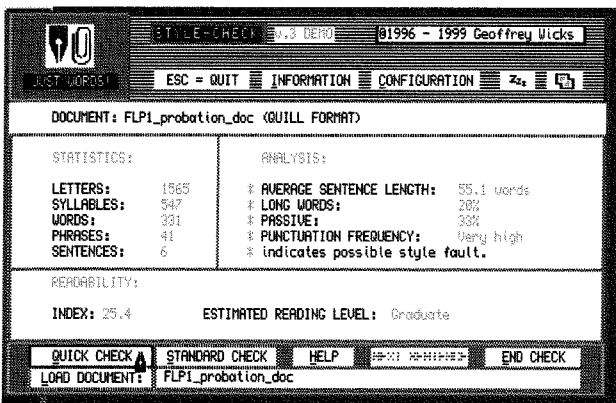
The program to be configured is: FLIGHT_OBJ
 You can specify the directory in which the nav aids_cde and world_cde files are stored (default=flp1_)

You can also specify the delay factor to be applied to the program. This is to help you use the program on fast QLs - standard QL's need a factor of 0 (zero) whereas Super Gold Card QLs need around 32768 (maximum value).

Please read the MANUAL_DOC for full instructions, which includes a list of all the keyboard controls etc, plus hints on how best to take off etc.

STYLE-CHECK DEMO VERSION

Author: Geoff Wicks
Price: £10 from Just Words
Documentation: Style_doc (Quill file)
Requirements: Pointer environment, expanded memory



This is the demo version of STYLE-CHECK, the only style checker for QDOS/SMSQ(/E) compatible computers. To use this program you will need disk drives and at least 512K memory. This demo version is a fully working version of STYLE-CHECK, but has 3 restrictions:

1: The demo version can be run only from flp1_. The full version can be run from any drive and can be installed on a Hard Disk.

2: Although a document of any size can be loaded into the program (provided your system has sufficient memory), the demo version of STYLE-CHECK will examine only the first 3,000 bytes. In most documents this will be equivalent to 250 - 350 words.

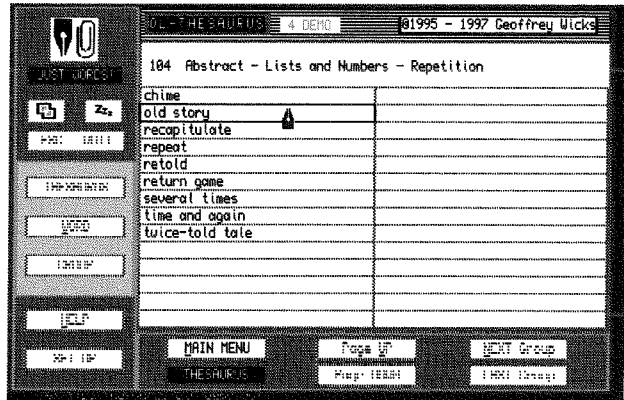
3: The demo version does not come with a printed manual. The 11,000 word manual supplied with the full version is an essential reference work for use with the program. In practice this restriction means that some of the comment messages you receive will be difficult to understand.

The best way to learn to use STYLE-CHECK and

discover its many possibilities is to follow the short tutorial in the supplied STYLE_DOC manual.

QL-THESAURUS - DEMO VERSION

Author: Geoff Wicks
Price: £10.00 from Just Words
Documentation: THESAURU_doc (Quill file)
Requirements: Pointer environment, Expanded memory



This is the demonstration version of QL-THESAURUS, which, as its name implies, is an electronic thesaurus for the QL and compatible computers. It is primarily designed to multitask with QL word processors to provide the user with fast on-screen help during a word processing session. QL-THESAURUS is heavily based on the classic classification of words devised by Roget. The full version of QL-THESAURUS has a data base of 23,000 words and phrases divided into 1,000 different groups.

This demonstration of QL-THESAURUS is a fully working version of the programme, but has two restrictions. Firstly the data base has only 4,600 words. A larger data base cannot be loaded into the demo version. The data base was compiled by selecting every fifth word or phrase from the data base of the full programme. Secondly the demo version can only be run from flp1_. The full version can be run from any drive and can be installed on a hard disk.

To run the full version of QL-THESAURUS you will need a QL or QDOS compatible system with disk drives and at least 640K memory. When fully loaded with its dictionary the full version requires about 490K. It would be difficult to multitask QL-THESAURUS with a word processor on anything less than a Trump Card. Toolkit 2 should be active and the pointer environment files installed. QL-THESAURUS is a true thesaurus and not just a synonym dictionary. In other words it is concerned with the ideas conveyed by words, whereas a synonym dictionary is simply concerned with meanings. As an example of the

difference, if you look up "Poisonous" in a synonym dictionary, you will find the words "Toxic" and "Venomous". You will also find these words in a Thesaurus, but in addition you will find "Snake", "Serpent" and "Scorpion".

In this program the usual PE conventions of Space = Hit or the left mouse button and Enter = Do or the right mouse button are followed. Please note that the Move command will work only with high resolution screens.

To explore the possibilities of QL-THESAURUS you should follow the short tutorial in the manual file THESAURU_DOC. Bear in mind that in the full version of the programme you will get about 5 times as many words in each category as in the demo version.

Text 87 Plus4 Demo Version

Author: Fred Toussi, Software 87
Price: £79.00 from QBranch
Documentation: DEMO_t91 (Text 87 file)
Requirments: Expanded memory

```

Cursor move: <+> <↓> <+> <→> plus <SHIFT> or <ALT> keys
Delete text: <CTRL> plus a cursor move key combination
0 Roman PS Sw
0 Normal
Help: <F1> Window: <F2> Commands: <F3> Typeface: <F4> Ruler: <F5>
Ruler 0 [Line spacing 30/180] [Justified] [.] [Selected] [0 0 180]
> . . . . .
PLUS4 DEMO

The enclosed disk contains a demo version of plus4. The plus4 demo
program is identical to the full version except that it will not save,
export or print and will not perform spell checking (unless you have
already installed the Spell Device from QJump). The demo version
contains only a demo printer driver based on the Epson LQ driver from
the 2488 series of drivers. The full version of plus4 and its extra
printer drivers also include a configuration program, several
additional screen display fonts, printer drivers, dictionaries and
ancillary files.

A tiny part of the 100 page manual is
reprinted here to get you started. The
full help file has also been included.

Text: demo Words: 3330 Line: 0 Section: 0 Page: 1 0:0

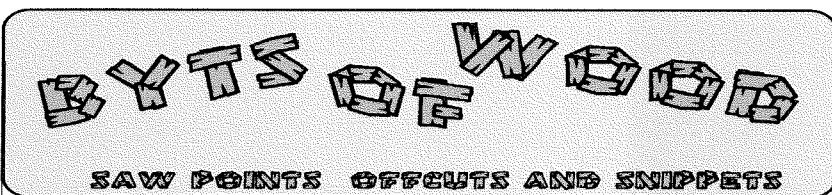
```

The Plus4 demo program is identical to the full version except that it will not save, export, or print files and will not perform spell checking (unless you have already installed the Spell Device from QJump).

The demo version contains only a demo printer driver based on the Epson LQ driver from the 2488 series of drivers. The full version of Plus4 and its extra printer drivers also include a configuration program, several additional screen display fonts, printer drivers, dictionaries and ancillary files.

A tiny part of the 100 page manual is included in the Demo_t91 file to get you started, and the full help file is also included.

To use the demo program, place the unzipped Plus4 Demo disk in FLP1_ and either LRUN FLP1_BOOT and follow the prompts, or enter the command EXEC FLP1_PLUS4DEMO. On a QL without pointer environment, after the prompt appears on the screen, hold down the CTRL key and tap the C key (i.e. use CTRL C) to move the flashing cursor to Plus 4. Press L for Load in the small menu which appears, then press cursor up or down to enter the file selection list. In the list, use the cursor up and down arrow keys to select the file called Demo_t91 in the list and press ENTER. Press ENTER again to load the file. Finally press ESC to read the text on the screen. Use the cursor up/down arrow keys to move through the text.



OK, before we start this month, I think I should conduct a quick straw poll. Hands up all those who don't know that I, Roy Wood, write this column. Ah very good that seems to be most of you. There are clues, of course, in the title and in the constant references to QBranch and trips to shows etc. so I am not too surprised you have all got the right answer. You don't really need

to see my name written down all the time do you?

I am also not sure I could be regarded as an 'officer' of this magazine. I do write this column, contribute the occasional article, do the UK distribution and draw the odd (very odd, some say) cartoon but that is a fraction of the work in producing this periodical.

I have said all of this because, although I do not think that

Peter Graf was referring to me in some of his remarks in the last issue, I wanted to state again, hopefully for the last time, that all views, ideas and comments come from me and are not the standpoint of the magazine itself. If you have any argument with my views or idea then please write in and put your point of view. I know that the editors would welcome a more active letters page.

I think that Peter was referring to the little bold type inserts which appear throughout the magazine. These are fairly common and usually occur

when one of the editors has a useful point to add that might clarify or correct something in the text. They are printed in Bold Type in order to differentiate them from the main text body. These have appeared in this column too (usually to correct one of my mistakes) and I, for one, have no problem with them. Maybe, in the light of Peter's objection, they should have initials added to the little pieces so we know which of the editors has made the comment.

As far as I can tell by talking to Dilwyn, Jochen and Bruce this magazine has no actual stance on any of the issues in the 'Big Debate'. Bruce has always had a large amount of native hardware and was one of the first to buy a Q 40 - He was also an early QPC 2 user. Dilwyn has often stated his preference for the old QL and his dislike of PCs and Jochen has, for a long time, run his systems on Ataris and QPC 2. I personally feel this is a fairly wide viewpoint and that the magazine is pretty unbiased because of it. This is one of the reasons I am very happy to keep supporting it. Still, if you can't win an argument with good logic you have to resort to crude innuendo - don't you?

If this magazine has a standpoint at all it is that of supporting QDOS/SMSQ Users whatever system they choose to run this on. I think that is a very admirable cause and I will rally behind that flag.

Oh and by the way, the two page adverts Q Branch has in each issue are paid for at the same rate as any other advertiser. Just so you don't think I have any unfair advantage or bias. I know that most of you knew this already but.....

Pumping Pixels

At the German / Austrian show in October I saw another great advantage for using native QL hardware. Both Rudolf Rindbacher and Wolfgang Ulrich carried in enormous 21" and 19" monitors to run their Systems. Obviously using these big things provides lots of exercise which I don't get when I carry in my laptop with QPC 2. Wolfgang was, admittedly, using an AMD driven PC to run QPC 2 but, for travelling around to do these far away shows you can't beat a tiny laptop - even though it means that the only exercise I get is lifting my wine glass after the show!

State of the Minute, Up to The Art Technology

Some time ago we embarked upon a discussion of the QL file system with various luminaries proposing changes to it in order to bring it into line with other systems in use with PCs, LINUX etc. At the time I vehemently opposed to changing this, not because I particularly loved the QL way, but because of the chaos it would cause older users and even those who, like myself, embrace most of the newer concepts while running some older software which could not be changed.

I have not changed my view in this matter but there is, of course, another side to this argument which we did not explore at the time. This came up in a conversation I had with Tony Firshman a few weeks ago. He had to upgrade his PC, not because he needed better performance, but because his son, Ben, had a program that would not run on his Pentium 1 233 system. The most economical way to give Ben a better system was to pass his one on

and get a better one himself.

Tony was bemoaning the fact that systems like this become out of date so quickly, whereas they were, not so long ago, 'state of the art'. The QL has soldiered on through many changes but still most of the software that we run is quite happy running on a minimal spec of QL, Gold Card, DD disks. This is, of course, a great advantage to the user but it has, in many ways, acted against the interests of hardware development.

Working, as I do for a day job, in the PC side of things I have noticed that the development of hardware is driven by software. New graphics chipsets, Processors and other peripherals are, of course, designed before the programs which use them but as soon as they are available the number of programs which have these as 'desired specifications' increases and, very soon, they become 'standard' and then on to 'minimum'. Users see a new game or application and want it but know that they cannot get it to run or, at least, get the best performance out of it on their current system so they upgrade.

This is something we have never done. Maybe the take up of the Aurora would have been faster if we had had more programs which used its capabilities to the full and either would not run or were severely curtailed on older systems. This is not just a commercial decision but an immensely practical one. The more people who upgrade to newer hardware, the better the programs can be. Programmers and designers stay motivated to keep working by the push for better systems and software. Apart from this the need to make software run on all platforms has led some pro-

grammers to rule out some features which would not be supported on older platforms. Problems like the need to fit software on DD disks also hold us back. It may not be too late to adopt a new approach, and Jim Hunkins comments at the US show this year are a step in the right direction.

I would welcome some feedback on this – especially from programmers or designers who feel these constrictions.

Are eXasPerated?

My day job also gives me the opportunity to play around with the latest hardware and operating systems. When Micro\$oft released their 'latest' operating system Windoze XP we set up a Pentium 4 PC and installed a copy on that. This was mostly to test the stuff we sell to make sure it would all run (even a month after it's release date there were still major manufacturers who had not released drivers for their products – if the mainstream cannot get this together what chance do we have with no access to any real information or even any help from the manufacturers themselves).

This allowed me to sneak in with my QPC2 CD. I made up a CD with QPC2 and a QXLWIN file on it so I could quickly test it out on various systems. Of course I have to reconfigure QPC2 to find the QXLWIN file each time because the systems do not always have D: as the CD ROM but it is a quick way to get access to my data files on any PC with Windoze and a CD ROM.

You will be pleased to know that, unlike some commercial software (AOL V7 to name but one) QPC2 fired up with no problems under Windoze XP

Licensed to Annoy

Slipping slightly off topic from QL stuff XP does make an interesting departure in tactics for Micro\$oft. Brought up, as we have been, in 'QL Mode' we are used to purchasing a copy of the software and using it on as many systems as we have. For some of us that is quite a few. QL Software has never gone in for protracted license agreements and these have never really been needed since the software was pretty cheap and the majority of users are very honest.

Micro\$oft, on the other hand, has always had a long, wordy piece of legalese to read through and requires that you accept it before allowing the installation of the software. How many of you have ever read this? How many of you realise that, even after forking over your money you do not own a copy of the software. What you in fact own is a license to use that software on a single machine. That license is, furthermore, not transferable so people who sell second hand computers with Windoze installed or even those who buy such machines are breaking the law and could be prosecuted for breach of copyright.

Micro\$oft have never made too much of this on the consumer level, preferring to vent their spleen on the volume pirates, the advent of the XP branch of O/S and Office software requires activation from Micro\$oft if the user wants to continue using it after 30 days. Although this activation does not require registration with Micro\$oft it does give them the upper hand in one particular area which could prove very lucrative.

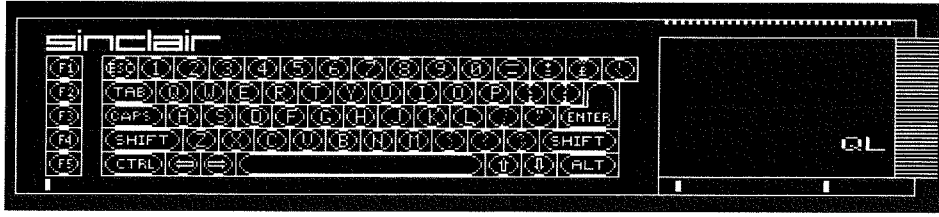
Many of the above second-hand systems which are sold via local papers and small computer shops run Windoze 95 or first edition 98. Once you have one of these systems and a master disk you need no further contact with Micro\$oft. You can effectively use it for as long as you want. The advent of Windoze 98 saw the withdrawal of support for 3.1 and XP saw a similar move for 95. Although the support has been withdrawn the systems still operate and can be re-installed if needed.

Under XP the system setup is recorded by Micro\$oft's registration utility. It knows what sound card, processor, motherboard etc you are using when the system is installed and that forms part of the code you have to give to Micro\$oft to activate the product. If you change enough stuff on the system then you have to re-activate it. If Micro\$oft decides to withdraw support for the system at any point all it has to do is to refuse re-activation. You are then faced with having to buy the latest version and part with more money. This is the creeping move towards the idea I mentioned in one of my earlier columns which they call 'NET'. You hire the software from them and pay a yearly fee to be able to use it. What is worse they can force you to upgrade to a newer system in this way – As they say in 'Oliver' 'You've got to pick a pocket or two...'

Looking over my Shoulder

Sometimes I am not too sure who is reading this column. Last issue I wrote about the floppy drive and how it had remained the most steadfast piece of 'heritage' hardware in

DILWYN JONES



QL P.D. SOFTWARE LIBRARY SERVICE

A brand new PD software library service from QL author Dilwyn Jones. Hundreds of freeware, shareware and PD programs available on floppy disk at a price of just £1.00 per disk (or just 75 pence if you supply the floppy disk). Programs are normally supplied on HD disks unless you specifically request DD disks.

CATALOGUE

To obtain a free catalogue of the entire software library:

1. Send me a formatted HD or DD disk and return postage, or
2. Download the catalogue from my website (see address below), or
3. By email – just send me an email to ask for the catalogue as a text file.

CD-ROMs

(Also available from Q-Celt Computing in Ireland)

QL EMULATORS CD **£5.00** - All of the available QL emulators!

QL PD-CDR **£5.00** - Gérard Plavec's software collection!

LINE DESIGN CLIPART CD **£10.00** - Huge collection of Line Design clipart!

FAMOUS FACES CD **£10.00** – Line Design pictures of the famous!

QL RELIGION CD **£10.00** – A collection of bibles, religious texts, clipart etc

DJ PD LIBRARY CD **£10.00** – The entire PD library on a CD! Start your own!

The CD-ROMs are supplied in QXL.WIN format on an ISO-9660 CD-R for use with QXL, QPC, QemuLator v2, uQLx, Q40, Q60 (systems able to read QXL.WIN media)

POSTAGE UK/Europe: add £1.00, Rest Of World: add £2.00.

PAYMENT In UK Pounds Sterling currency. Cheques payable to DILWYN JONES.

Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT, U.K.

Email: dilwyn.jones@dj.softnet.co.uk

Website: <http://www.soft.net.uk/dj/index.html>

Note: Although this service is run by the editor of QL Today magazine, it is a completely separate venture which has no connection with the publisher of this magazine. Software is supplied free of charge, apart from copying and media charges etc to cover costs, and without any warranty other than to the replacement of defective media (e.g. "bad or changed medium" errors etc.). In other words, usual PD library terms. E & OE.

the PC stable. I also commented that the connection and configuration of this device were still in the stone age.

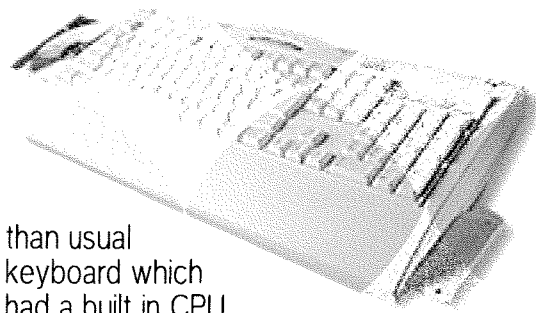
No sooner had this copy gone to Jochen but 'The Register' reported that IBM were going to drop 3.5" drive support from the next generation of systems. Not only this but they were planning to drop the serial, PS/2 and parallel ports as well.

Moves like these are the downside of comments mentioned above where programmers take hold of the better hardware and write programs to inspire people to buy it. The trouble is that they are two sides of the same coin. Development cannot take place without investment and investment needs a return.

Shortly after this they released a statement saying that they had no such plans but it all smacks of the usual way that the computer industry tries to lead the consumer by the nose with its hand firmly dipping into the unsuspecting mug's pocket while it does so. Watch out for more of this kind of activity as the belts tighten in the wake of the current downturn in the consumer electronics market. I have probably said this before but buying a new PC makes buying a brand new car look like a good investment. One good reason to stay with your old QL.

Sounds Familiar

In one of the magazines that we get at work I noticed an ad for a system trumpeting its 'new concept'. It proudly states that the old idea of a bulky tower case with a big desktop 'footprint' (probably where the user kicked it in eXasPeration) are numbered. This ad had a picture of the new system with a slightly thicker



than usual keyboard which had a built in CPU, Hard Drive, Floppy Drive, CD ROM and all of the outputs along the back. The system was made by a company called PAYSAN and called Cybernet Elite II. Now take the above and remove the CD-ROM, spray it black and you will have something very similar to the QLs modified by Keith Mitchell and Roger Godley. Well there's nothing new really is there ?

More Ink Less Control

A while ago Jochen and I became very interested in the EPSON Stylus 880 printer which used the ESC/P2 codes and worked very well with the QL. It was fast, quiet and was available at a good price. Whoopee! we thought. We joked at one of the recent shows that the printer would be discontinued because it was 'too good to sell' and - guess what? EPSON have made it obsolete.

Of course this happened just as I had decided to supply them to QL Users so there were a few people who were disappointed. The new range of printers that EPSON brought out seemed interesting and I embarked upon a quest to find out if any of them had the required ESC/P2 interface. New listeners read on.....

Standards are Just Things that Flap in the Breeze - Part II

Well EPSON had a printer in the range called the C80 which

looked promising. First of all it boasted the ESC/P2 driver, secondly it seemed to have a very good spec of 2880/2880 dpi and lastly it had separate ink tanks for all of the colours - great for people who

print out a lot of red letters I thought.

OK, I thought, I'll take the plunge and bought one, hoping to be able to produce a good review for the magazine.

To start with everything looked fine. I set it up on the PC and it produced good quality colour print with no problem. This, however, was not the end of the story, as you can guess, and I will save the full story for a later review. Suffice it to say that the printer will produce good colour on my PC but only Black and White on any of the QL applications. It would seem that EPSON have changed the ESC/P2 standard without telling anyone. I am still working on this. Watch this space.

It has, however, led me to discover the reasons behind some problems I have had with printers for some time. I would like to relate a little story of printers past.

The Strange Saga of the 740

When I lived in Germany my wife and I shared an office space and an EPSON Stylus 800 printer. This was connected via a switch to both her PC and my QL. Then we moved to the UK and Q Branch started up. Q Branch Towers has a converted loft space and that became my office (she hated the stairs to it) so I ran a printer cable through the ceiling and the switch arrangement continued along with a new com-

munication method whereby she would stand at the bottom of the stairs and shout up 'Can I print?' and I would turn the switch.

Printers came down in price and I finally bought an EPSON Stylus Colour 200 passing the 800 on to her. By this time there were two systems in Q Branch Towers, a tower cased Aurora and my laptop running QPC - the switch was still doing sterling service. The 200 had one drawback and that was it required the replacement of it's print head whenever you wanted to change from black to coloured ink. A couple of years ago Jochen got hold of a very cheap EPSON Stylus 740 for me so I decided to retire the 200 to Workshop duty and install that. It refused to print at all from the QL system even though it was one of the few printers that supported ESC/P2 and many others had used it on QL systems. Two things had happened by this time. My laptop had been superseded by a desktop PC and the QL system had been augmented by the Q 40 which was just beginning to be usable. I moved the 740 to the PC where it printed fine and put the switch between the Q 40 and the MinisQL (the tower case had become a MinisQL) and everything settled down. Bill Waugh sold me a Stylus 500 which took over from the 200 and had the advantage of colour and Black ink at the same time and the 200 became the show machine. The 740 would not print from QPC 2 but I assumed that was down to a fault on the 740 and, anyway, I did most of my work on the Q 40.

Now we come to the present day and the C80. The MinisQL was now being used only at

shows for disk duplication and the Q 40 was doing all of the QL stuff very well. The switch box had been retired. I brought the C80 home and plugged it into the PC as above. Fine - it works and that was the computer I wanted it for. It still did not print from QPC 2. Now this was odd because I had an HP Laserjet 3 from the old Bank Volt which was, for a time connected by the switch to the PC and that printed fine. I began to expect that it would not work with the Q 40 either. As I said above, it printed but on in Black. I then plugged the 740 in (I don't know why) and that printed perfectly well in colour. Arrrrrrggggggg!

While talking to Jochen on the phone about this I happened to mention the switch and told me that the 740 had given him, problems with a switch as well and it was something to do with the way that the switch added resistors to the data cables or made the total length of the parallel port cable simply too long. I wish someone had mentioned that before. I then wondered about why it would not print from QPC 2. The C80 will not print from this either. At the moment I have no firm answers but I suspect it has something to do with the EPSON status monitor software which may not work very well with QPC 2 burying it. I hope to have a lot more news of this in the next column or even a review for the next issue.

And Now

Christmas Entertainment

Many of you enjoyed my round up of Christmas entertainment for computer users so I have a similar one for this year. (If you don't get them at first try

saying them out loud). First off a few films.

Some computer action films:

**The QWERTY Dozen
Shutdown at the Click OK
Corral
Sink The Benchmark!
Dilwyn Jones and the RAM
Disk of the Lost Ark
The Firsh Man on the Moon
Crash!
Lockup, Stuck and Two
Smoking Cables
Honeyball (follow up to
Silence of the LANs)**

Some Classic Period Dramas:

**The Rise and Fall of the ROM
and GoldFire (now gone
Platinum)
Goodbye All My Chips**

The Obligatory Agatha Christie
Blockbuster:

**(Its) Murder Using Outlook
Express**

And the even more Obligatory
Bond Movies:

**The Man With The Gold Card
ROM (featuring Minnie
Driver's younger sister
Micro)
QL World Is Not Enough**

And for the Kids:

**Fontasia
Lady and the Trump Card**

Lastly for Opera lovers there
are some digital Operas:

**Al Eda
CPU Fan Tutte
Turan.com**

This all leaves me to
wish you all a happy
Christmas and a very
QL New Year!



The QL Show Agenda



Hove Workshop - (UK)

3rd March 2002 at Portslade Town Hall on the south coast (same venue as last time).
Good parking and easy access from Portslade station.
Contact Roy Wood at QBranch for further details.

Quanta AGM and Workshop - (UK)

Manchester. A 2-day event on Saturday 13th and 14th April 2002,
Venue: 3rd Davyhulme Scout Headquarters, Conway Road, off Lostock Road,
Davyhulme Manchester; just 2 minutes from junction 9 M60
Times: Traders from 12noon Sat to 6pm and Sun 9.30 to whenever
Public from 2pm Sat and from 10am Sun. AGM 2.00pm Sunday
Accommodation: Travel Inn at the Trafford Centre, off Junction 10, M60
Travel Inn at Mersey Farm, end of Carrington Spur, Junction 8, M60
Theme: QL – The Way Ahead or The 21st Century, here we come!
Dinner Sat evening: Numbers required by end of Feb 2002, with £5 deposit please
Speakers: invited on that theme, please. More news next issue.



Not too many QL related events to report this time I'm afraid, but as you'll see from this issue there's plenty happening to keep QLers busy!

We would like to extend our sincere thanks to our authors and subscribers who kept QL Today going during its sixth year and hopefully well into the future. Hopefully, with your help, we'll be able to bring lots more interesting QL articles in 2002. If you don't find what you're looking for in QL Today, set pen to paper or finger to keyboard and write an article yourself to encourage others to do the same. We all benefit by each other's efforts! We would like to wish all our readers a merry christmas season and a very happy new year to you all.

